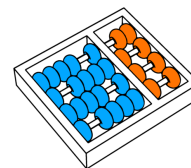


Neumar Costa Malheiros

“Uma Abordagem Cognitiva para Auto-Configuração de Protocolos de Comunicação”

CAMPINAS
2013



Universidade Estadual de Campinas
Instituto de Computação

Neumar Costa Malheiros

“Uma Abordagem Cognitiva para Auto-Configuração de Protocolos de Comunicação”

Orientador(a): **Prof. Dr. Edmundo Roberto Mauro Madeira**

Co-Orientador(a): **Prof. Dr. Nelson Luís Saldanha da Fonseca**

Tese de Doutorado apresentada ao Programa de
Pós-Graduação em Ciência da Computação do Instituto de Computação da
Universidade Estadual de Campinas para obtenção do título de Doutor em
Ciência da Computação.

ESTE EXEMPLAR CORRESPONDE À VERSÃO
FINAL DA TESE DEFENDIDA POR NEUMAR
COSTA MALHEIROS, SOB ORIENTAÇÃO DE
PROF. DR. EDMUNDO ROBERTO MAURO
MADEIRA.

Assinatura do Orientador(a)

CAMPINAS
2013

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

M294a Malheiros, Neumar Costa, 1981-
Uma abordagem cognitiva para auto-configuração de protocolos de comunicação / Neumar Costa Malheiros. – Campinas, SP : [s.n.], 2013.

Orientador: Edmundo Roberto Mauro Madeira.
Coorientador: Nelson Luis Saldanha da Fonseca.
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Redes de computadores - Gerência. 2. Redes de computadores - Protocolos. I. Madeira, Edmundo Roberto Mauro, 1958-. II. Fonseca, Nelson Luis Saldanha da, 1961-. III. Universidade Estadual de Campinas. Instituto de Computação. IV. Título.

Informações para Biblioteca Digital

Título em inglês: A cognitive approach to self-configuration of communication protocols

Palavras-chave em inglês:

Computer networks - Management

Computer network protocols

Área de concentração: Ciência da Computação

Titulação: Doutor em Ciência da Computação

Banca examinadora:

Edmundo Roberto Mauro Madeira [Orientador]

Célio Vinicius Neves de Albuquerque

Lisandro Zambenedetti Granville

Islene Calciolari Garcia

Juliana Freitag Borin

Data de defesa: 16-05-2013

Programa de Pós-Graduação: Ciência da Computação

TERMO DE APROVAÇÃO

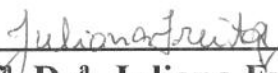
Tese Defendida e Aprovada em 16 de Maio de 2013, pela Banca
examinadora composta pelos Professores Doutores:



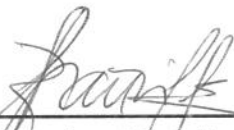
Prof. Dr. Célio Vinicius Neves de Albuquerque
IC / UFF



Prof^a. Dr^a. Islene Calciolari Garcia
IC / UNICAMP



Prof^a. Dr^a. Juliana Freitag Borin
IC / UNICAMP



Prof. Dr. Lisandro Zambenedetti Granville
I-INFO / UFRGS



Prof. Dr. Edmundo Roberto Mauro Madeira
IC / UNICAMP

Uma Abordagem Cognitiva para Auto-Configuração de Protocolos de Comunicação

Neumar Costa Malheiros¹

16 de maio de 2013

Banca Examinadora:

- Prof. Dr. Edmundo Roberto Mauro Madeira (*Orientador*)
- Prof. Dr. Célio Vinicius Neves de Albuquerque
Instituto de Computação – UFF
- Prof. Dr. Lisandro Zambenedetti Granville
Instituto de Informática – UFRGS
- Profa. Dra. Islene Calciolari Garcia
Instituto de Computação – UNICAMP
- Profa. Dra. Juliana Freitag Borin
Instituto de Computação – UNICAMP

¹Suporte financeiro de: Bolsa de Doutorado da FAPESP (processo 2006/50512-4) 2006–2009 e 2010–2011 e Bolsa de Doutorado Sanduíche do Programa Erasmus Mundus (ECW-ISAC) 2009-2010.

Abstract

As network technologies evolve, the complexity of managing communication infrastructures and protocols increases. Such complexity makes the management of current communication networks a major challenge. Traditional centralized solutions for network management are not scalable and are incapable of providing continuous reconfiguration of network protocols in response to time-varying conditions.

In this work, we present a feasible and effective solution for self-configuration of communication protocols. We propose a cognitive approach for dynamic reconfiguration of protocol parameters in order to avoid performance degradation as a consequence of changing network conditions. The proposed cognitive framework, called CogProt, provides runtime adjustment of protocol parameters through learning and reasoning mechanisms. It periodically reconfigures the parameters of interest based on acquired knowledge to improve system-wide performance. The proposed approach is decentralized and can be applied to runtime adjustment of a wide range of protocol parameters at different layers of the protocol stack.

We present a number of case studies to illustrate the application of the proposed approach. Both simulation and wide-area network experiments were performed to evaluate the performance of the proposed approach. Results demonstrate the effectiveness of the proposed approach to improve overall performance for different network scenarios and also to avoid performance degradation by timely reacting to network changes.

Resumo

À medida que novas tecnologias de rede são desenvolvidas, torna-se mais complexa a tarefa de gerenciar os serviços e protocolos de comunicação. Diante de tal complexidade, a gerência das redes de comunicação atuais representa um grande desafio. As soluções de gerência tradicionais, com uma arquitetura centralizada, não apresentam um alto grau de escalabilidade e não são capazes de prover reconfiguração dinâmica dos protocolos de comunicação em resposta às constantes mudanças nas condições da rede.

Neste trabalho, apresenta-se uma solução factível e eficaz para auto-configuração de protocolos de comunicação. Propõe-se uma abordagem cognitiva para a reconfiguração dinâmica de parâmetros de protocolos a fim de evitar a degradação de desempenho resultante de eventuais alterações nas condições da rede. O arcabouço proposto, denominado CogProt, provê, através de mecanismos de aprendizagem e decisão, o ajuste de parâmetros dos protocolos durante a operação da rede. Parâmetros de interesse são periodicamente reconfigurados de acordo com informações de monitoramento a fim de aumentar o desempenho médio do sistema como um todo. A abordagem proposta é descentralizada e pode ser aplicada no ajuste dinâmico de uma ampla variedade de protocolos em diferentes camadas da arquitetura da rede.

Apresenta-se uma série de estudos de caso para ilustrar a aplicação da abordagem proposta. Estudos baseados em simulação e um experimento em um ambiente de rede real foram realizados para avaliar o desempenho do arcabouço CogProt. Os resultados demonstram a eficácia da abordagem proposta em reagir prontamente às mudanças no estado da rede e melhorar o desempenho médio dos protocolos.

Sumário

Abstract	ix
Resumo	xi
1 Introdução	1
2 Gerência Autônoma de Redes	5
2.1 Abordagem Tradicional	6
2.2 Gerência Baseada em Políticas	7
2.3 Gerência Autônoma	9
2.3.1 Propriedades de Auto-Gerência	9
2.3.2 Laço de Controle	11
2.3.3 Desafios em Redes Autônomicas	12
2.4 Considerações Finais	13
3 Trabalhos Relacionados	14
3.1 Abordagens de Auto-Configuração	14
3.2 Comparação Qualitativa	16
4 CogProt	18
4.1 Visão Geral da Abordagem Cognitiva	19
4.2 O Processo de Auto-Configuração	20
4.2.1 Análise	21
4.2.2 Decisão	24
4.2.3 Ação	25
4.3 Descrição do Processo como Algoritmo	26
4.4 Exemplo	28
4.5 Cenários de Aplicação	31

5	Estudos de Caso	35
5.1	Auto-Configuração do TCP	36
5.2	Adaptação da Taxa de Transmissão	41
5.3	Múltiplos Parâmetros	46
5.4	Implementação em Ambiente Real	50
6	Conclusão	54
6.1	Trabalhos Futuros	56
	Referências Bibliográficas	57
A	Implementação do CogTCP	62
A.1	Código Fonte do CogTCP	63
A.2	Código Fonte do Makefile	67

Lista de Tabelas

3.1	Comparação entre arquiteturas de redes cognitivas.	16
4.1	Definição do parâmetro de interesse.	26
4.2	Parâmetros de controle do processo de auto-configuração.	26
4.3	Exemplos de parâmetros para aplicação do CogProt.	31

Lista de Figuras

2.1	Arquitetura tradicional de sistemas de gerência com uso do protocolo SNMP.	6
2.2	Arquitetura da abordagem de Gerência Baseada em Políticas.	8
2.3	Rede formada por elementos autônômicos.	10
2.4	Arquitetura de um elemento autônômico.	11
4.1	Visão geral da abordagem cognitiva.	19
4.2	Arquitetura de um elemento de rede cognitivo.	21
4.3	Etapas do processo de auto-configuração.	22
4.4	Monitoramento de desempenho na etapa de análise.	23
4.5	Ajuste da média μ na etapa de decisão.	24
4.6	Ajuste do parâmetro na etapa de ação.	25
4.7	Função densidade para distribuição normal padrão.	25
4.8	Exemplo numérico.	30
4.9	Cenários de auto-configuração de múltiplos parâmetros.	32
4.10	Arquitetura centralizada (baseada em servidor).	33
4.11	Arquitetura descentralizada.	34
5.1	Cenário de rede simulado na auto-configuração do TCP.	38
5.2	Vazão média das conexões TCP para valores fixos de α e para o CogTCP.	39
5.3	Vazão média agregada para valores fixos de α e para o CogTCP em um cenário heterogêneo.	40
5.4	Vazão instantânea ao longo da simulação para CogTCP, $\alpha = 1$ e $\alpha = 2$ em um cenário heterogêneo.	40
5.5	Cenário de rede simulado para avaliação do CORA.	42
5.6	Vazão média para as taxas 802.11g.	43
5.7	Vazão em função do tempo.	44
5.8	Vazão média dos algoritmos.	44
5.9	Cenário de rede simulado para avaliação do CORA.	45
5.10	Vazão média.	45

5.11	Cenário de rede para avaliação da abordagem cognitiva no ajuste dinâmico de múltiplos parâmetros.	46
5.12	Vazão média sem tráfego de interferência para os casos onde S está próximo ou distante de BS	47
5.13	Vazão média com a presença de tráfego de interferência.	48
5.14	Vazão média em um cenário heterogêneo.	49
5.15	Cenário de rede do experimento com CogTCP.	50
5.16	Desempenho nos diferentes cenários de rede.	52
5.17	Desempenho médio com fluxos nas duas redes.	53
5.18	Frequência de escolha dos valores de α	53

Capítulo 1

Introdução

As redes de comunicação atuais são sistemas com um alto grau de complexidade. Muitos fatores contribuem para isso. Em primeiro lugar, pode-se destacar a heterogeneidade das tecnologias de transmissão e dos próprios elementos de rede. Por exemplo, os equipamentos que podem acessar a Internet incluem desde dispositivos portáteis até supercomputadores. Outro fator crítico é o suporte à mobilidade dos usuários. O desenvolvimento de redes de acesso sem fio de alto desempenho ainda é um grande desafio para pesquisadores na academia e na indústria. Por fim, um terceiro fator consiste nos requisitos de QoS (*Quality of Service*) e QoE (*Quality of Experience*) [21] de aplicações científicas e aplicações multimídia avançadas. A plena garantia de níveis de serviço ainda não é uma realidade em redes de dados de larga escala.

Toda essa complexidade reflete em diversos aspectos do projeto e operação de redes. Em particular, a gerência de infraestruturas de comunicação envolve processos muito complicados e dispendiosos para operadoras de telecomunicação e grandes organizações. À medida em que são desenvolvidas novas tecnologias e serviços de rede, torna-se mais desafiador o projeto de soluções de gerência que sejam efetivamente capazes de prover o uso eficiente dos recursos. Historicamente, os primeiros mecanismos de gerência eram implementados seguindo um modelo cliente-servidor. Esses mecanismos consistiam em simples ferramentas de linha de comando, através das quais os administradores configuravam os elementos de rede. A introdução do protocolo SNMP (*Simple Network Management Protocol*) [11] permitiu o desenvolvimento de sistemas de gerência mais sofisticados. Entretanto, esses sistemas ainda seguiam uma arquitetura centralizada. Outro problema dessas soluções era a dependência humana do bom funcionamento da rede, uma vez que os administradores são os responsáveis por monitorar a configuração e desempenho do sistema e, então, disparar ações de gerência conforme necessário. Estas atividades realizadas manualmente aumentam a probabilidade de erros e tornam os processos mais lentos. De fato, as abordagens de gerência tradicionais não são adequadas para lidar com a crescente

heterogeneidade e dinâmica das infraestruturas de rede atuais [7, 32]. Há mesmo quem pondere que a ausência de mecanismos de controle e gerência eficazes é um dos principais entraves para avanços mais rápidos nos sistemas de comunicação [48].

A necessidade de soluções descentralizadas e com maior grau de automatização dos processos motivou o projeto de uma nova abordagem de gerência denominada Gerência Baseada em Políticas [53]. Nesta abordagem, ações de controle podem ser tomadas automaticamente, de acordo com políticas administrativas, para que elementos e serviços de rede possam reagir às mudanças na rede. As políticas são conjuntos de regras, com alto nível de abstração, definidas a partir de objetivos administrativos da rede. Essa abordagem reconhecidamente promoveu avanços e foi usada em diversas aplicações de gerência, principalmente gerência de segurança e de configuração. No entanto, ela apresenta algumas limitações entre as quais pode-se destacar a dificuldade na representação de políticas e na resolução de conflitos entre políticas. Além disso, apesar de prover aos elementos de rede uma certa capacidade de reagir às alterações nas condições da rede, as políticas possuem uma natureza estática, uma vez que representam conjuntos de regras previamente definidas. Diante das limitações das abordagens de gerência, foi proposto, recentemente, o conceito de Redes Autônomicas [10], que são redes capazes de se auto-gerenciar. Esse conceito foi definido de acordo com os princípios do paradigma de Computação Autônômica [37]. Em linhas gerais, pode-se definir um sistema autônomo como aquele capaz de gerenciar a si mesmo. O objetivo principal é que o sistema seja capaz de se adaptar. Neste contexto, as redes autônomicas são redes com mecanismos de auto-gerência que controlam o comportamento da rede e seus serviços, a fim de otimizar o desempenho do sistema como um todo.

Embora tenham sido definidas as propriedades de uma rede autônoma, ainda é um grande desafio o projeto de mecanismos de auto-gerência que possam efetivamente prover essas propriedades aos elementos da rede. As abordagens para auto-gerência descritas na literatura possuem algumas limitações [20, 31], entre as quais pode-se citar: mecanismos com uma arquitetura centralizada ou que necessitam de uma visão global da rede, o que compromete a escalabilidade da solução; mecanismos muito complexos para serem implementados em equipamentos com recursos limitados; mecanismos que não podem ser aplicados de forma genérica pois foram projetados para arquiteturas ou protocolos específicos.

Neste contexto, o objetivo deste trabalho de pesquisa foi investigar uma solução de gerência automatizada e descentralizada, para lidar com a dinâmica e a heterogeneidade das redes atuais. A principal contribuição deste trabalho é uma nova abordagem para a reconfiguração dinâmica de protocolos de comunicação, a fim de se evitar uma eventual degradação de desempenho decorrente de mudanças na rede. A abordagem proposta é denominada cognitiva porque envolve tarefas de aprendizagem e decisão. Em linhas

gerais, a gerência de redes pode ser dividida em cinco áreas: gerência de desempenho, de falhas, de configuração, de contabilização e de segurança. Neste trabalho, o foco é na gerência de configuração e desempenho. Não consideramos outras áreas como segurança ou mesmo gerência de mobilidade.

A abordagem proposta consiste em um arcabouço, denominado CogProt, para o projeto de mecanismos de auto-configuração de protocolos de rede. Esta abordagem foi inspirada no conceito de Redes Cognitivas [52, 23], que visa o projeto de mecanismos de aprendizagem e decisão para otimizar o desempenho do sistema de comunicação como um todo. O principal componente do CogProt é um ciclo de controle para o ajuste dinâmico de parâmetros de configuração dos protocolos. São descritas as etapas do ciclo de controle, assim como um algoritmo genérico para sua implementação. Apresenta-se, também, um exemplo para ilustrar o funcionamento do ciclo de controle. O arcabouço CogProt possui uma arquitetura descentralizada. Além disso, ele não depende de arquiteturas ou tecnologias específicas e não exige alterações nas mensagens ou no modo de operação dos protocolos. A abordagem proposta pode ser aplicada na auto-configuração de uma variedade de protocolos. São discutidos diferentes cenários de aplicação e os fatores que devem ser definidos para aplicações específicas do arcabouço. As características da abordagem proposta são comparadas com trabalhos relacionados. São apresentados, também, estudos de caso que ilustram a aplicação da solução proposta na auto-configuração de protocolos das camadas de transporte e de enlace. A solução proposta nesta tese foi usada no projeto de um mecanismo de auto-configuração do protocolo TCP (*Transmission Control Protocol*) e de um mecanismo de controle da taxa de transmissão em redes WiFi. Este último foi uma atividade desenvolvida em conjunto com o trabalho de mestrado de Luciano Chaves [15]. Além disso, foi realizado um estudo no qual parâmetros de ambas as camadas são simultaneamente ajustados. Além dessas avaliações realizadas por meio de simulação, foi feito um estudo de caso através da implementação do arcabouço em um cenário real. O mecanismo de auto-configuração do TCP foi implementado como um módulo do Linux usando o ciclo de controle proposto. Os resultados deste experimento foram coerentes com aqueles obtidos no estudo da auto-configuração do TCP através de simulação. Em suma, as avaliações realizadas ilustram a viabilidade e eficácia da abordagem proposta no projeto de mecanismos de auto-configuração que permitiram melhorar o desempenho médio dos protocolos.

Esta tese está organizada da seguinte forma. A seguir, no Capítulo 2, é apresentado um breve histórico da evolução das abordagens de gerência, com foco no conceito de Redes Autônomicas. No Capítulo 3, são discutidos trabalhos relacionados. São relacionadas importantes contribuições para arquiteturas de redes autônomicas e mecanismos de auto-configuração. Além disso, é feita uma comparação qualitativa entre a solução proposta e os trabalhos relacionados. A abordagem de gerência proposta é descrita no Capítulo 4.

Primeiro, é apresentada uma visão geral da proposta. A seguir, é descrito o elemento central da presente proposta, o arcabouço CogProt, que permite o projeto de mecanismos de auto-configuração. O processo de reconfiguração dinâmica é descrito em detalhes, assim como um algoritmo para sua implementação. Além disso, é apresentado um exemplo para ilustrar o funcionamento do processo de reconfiguração e são discutidos alguns cenários de aplicação do CogProt, considerando-se diferentes arquiteturas. Em seguida, no Capítulo 5, são apresentados os estudos de caso que foram realizados para avaliação da abordagem proposta. É apresentada a aplicação do CogProt na auto-configuração do protocolo TCP e no controle da taxa de transmissão em redes sem fio 802.11. Além disso, são descritos os resultados obtidos em um cenário de rede real a partir da implementação do mecanismo de auto-configuração do TCP. Por fim, no Capítulo 6, as conclusões são derivadas e trabalhos futuros são indicados.

Capítulo 2

Gerência Autônômica de Redes

A gerência de redes consiste em um conjunto de mecanismos que visam prover eficiência no uso dos recursos da infraestrutura de comunicação e, assim, otimizar o desempenho dos serviços de transmissão de dados com algum nível de segurança. Em linhas gerais, a gerência de redes pode ser dividida em cinco áreas: gerência de configuração, de desempenho, de falhas, de contabilização e de segurança. As áreas de gerência de configuração e desempenho estão intrinsecamente relacionadas, uma vez que o controle de desempenho da rede envolve a reconfiguração dos seus elementos de *hardware* e *software*.

Na prática, um sistema de gerência de redes deve suportar três tarefas principais [40]:

- i) *monitoramento* das condições dos elementos e serviços de rede para que se possa verificar se estão de acordo com o esperado;
- ii) *controle reativo* para ajuste de configuração do sistema diante de alterações nas condições da rede;
- iii) *detecção proativa* de anomalias (em relação a desempenho, falhas ou segurança) a fim de que ações de controle possam ser tomadas antecipadamente para se evitar problemas.

Neste capítulo, é apresentado um breve histórico da evolução das principais abordagens para o projeto de sistema de gerência. Primeiro, é descrita a abordagem tradicional, baseada no protocolo SNMP. Em seguida, são apresentados os avanços com a introdução da abordagem de Gerência Baseada em Políticas. Por fim, são apresentadas a abordagem de Gerência Autônômica, as principais propriedades de redes autônômicas e alguns dos desafios no projeto dessas redes.

2.1 Abordagem Tradicional

Os primeiros sistemas de gerência eram implementados seguindo um modelo cliente-servidor. Eles consistiam em simples ferramentas de linha de comando, através das quais os administradores configuravam os elementos de rede. Tais sistemas automatizavam apenas a tarefa de monitoramento. As tarefas mais complexas, de controle reativo e detecção proativa, ainda eram realizadas manualmente pelos administradores.

O projeto do protocolo SNMP (*Simple Network Management Protocol*) [11] permitiu o desenvolvimento de sistemas de gerência mais sofisticados. A arquitetura tradicional de sistemas de gerência de redes IP é baseada nesse protocolo. Essa arquitetura inclui três componentes principais: a entidade gerenciadora, o protocolo de gerência e o agente de gerenciamento. Na Figura 2.1, é ilustrado uma forma de como esses componentes poderiam estar organizados, considerando-se o uso do SNMP como o protocolo par compartilhamento de informações de gerência.

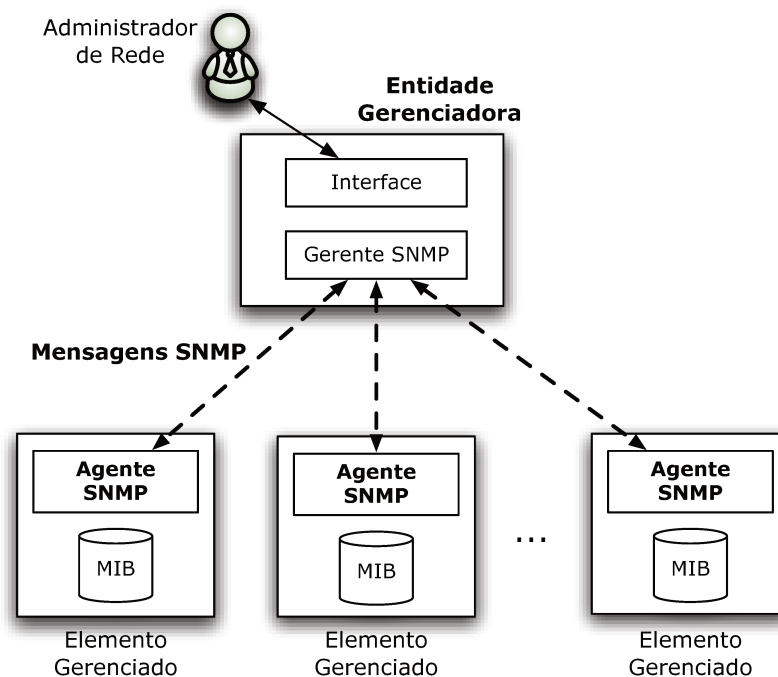


Figura 2.1: Arquitetura tradicional de sistemas de gerência com uso do protocolo SNMP.

Em geral, a entidade gerenciadora é denominada Sistema de Gerência de Rede (*Network Management System* – NMS). Esse sistema comunica-se com os agentes de gerenciamento, implementados nos elementos de rede, através do protocolo de gerência SNMP. Por meio das mensagens SNMP, a entidade gerenciadora pode consultar ou modificar parâmetros

de configuração nos elementos gerenciados. Adicionalmente, o protocolo permite que os agentes notifiquem a entidade gerenciadora sobre a ocorrência de eventos nos elementos gerenciados. Pode-se observar que, na verdade, o protocolo de gerência SNMP não gerencia a rede, ele apenas fornece serviço de compartilhamento de informações de gerência para que os administradores possam fazê-lo [40]. Além disso, é importante ressaltar que o uso do protocolo SNMP não requer, necessariamente, uma estrutura centralizada para o sistema de gerência. Sistemas descentralizados podem também usar esse protocolo para troca de informações de gerência.

As informações associadas a cada elemento gerenciado (por exemplo, valores de parâmetros de configuração) são armazenadas localmente em uma base de dados com as informações de gerência e controle. Essas informações são organizadas conforme regras de estrutura e sintaxe bem definidas de acordo com um modelo de dados conceitual denominado MIB (*Management Information Base*). Quando um administrador da rede realiza uma operação de consulta no sistema de gerência, o agente SNMP acessa a base do elemento gerenciado para retornar o valor correspondente.

2.2 Gerência Baseada em Políticas

Apesar de ter sido proposto há décadas, o protocolo SNMP ainda é amplamente utilizado em ferramentas de gerência modernas de grandes fabricantes de equipamentos de rede, como no caso do sistema CiscoWorks NCM [51]. O problema é que essas ferramentas ainda se caracterizam por uma arquitetura centralizada. Outra limitação é a dependência humana para operação da rede, uma vez que o sistema permite o monitoramento e reconfiguração dos elementos gerenciados, mas os administradores ainda são os responsáveis por tomar as decisões de gerência. Estas atividades realizadas manualmente aumentam a probabilidade de erros e tornam os processos mais lentos.

A necessidade de soluções descentralizadas e com maior grau de automatização dos processos motivou o projeto de novas abordagens de gerência, entre as quais pode-se destacar: Gerência por Delegação (*Management by Delegation* – MbD) [22, 24], Gerência P2P (*P2P-Based Network Management*) [5, 45], Gerência baseada em Agentes [4] e Gerência Baseada em Políticas (*Policy Based Management* – PBM) [8, 53].

A abordagem de Gerência Baseada em Políticas (PBM), além de permitir uma arquitetura descentralizada, destaca-se pelo foco em proporcionar um maior grau de abstração dos processos de gerência. O objetivo da abordagem PBM é a automatização de decisões de gerência de acordo com políticas pré-definidas. Essas políticas são conjuntos de regras, com alto nível de abstração, definidas a partir de objetivos administrativos e regras de negócio. Essa políticas devem permitir ao sistema de gerência decidir sobre ações de controle para reconfiguração de elementos e serviços de rede. Assim, o sistema poderia reagir

às mudanças nas condições da rede sem a necessidade de intervenção de administradores.

A arquitetura de um sistema de gerência baseado na abordagem PBM envolve quatro componentes principais: o repositório de políticas, o módulo de gerência de políticas, o módulo de decisão (*Policy Decision Point* – PDP) e o módulo de aplicação de políticas (*Policy Enforcement Point* – PEP). Na Figura 2.2, apresenta-se uma exemplo de como esses componentes podem se relacionar. Um sistema de gerência PBM pode incluir várias instâncias de cada um desses componentes de acordo com os requisitos de escalabilidade e desempenho do sistema. Por exemplo, um sistema pode incluir vários PDPs ou repositórios de políticas distribuídos. Além disso, o elemento gerenciado pode ser qualquer componente de *software* ou *hardware* da rede, assim, vários elementos gerenciados e seus respectivos PEPs podem estar em um mesmo elemento de rede.

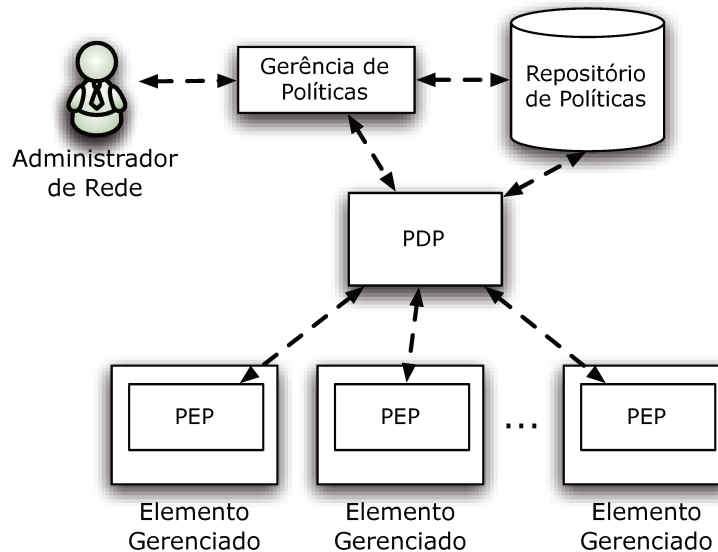


Figura 2.2: Arquitetura da abordagem de Gerência Baseada em Políticas.

As políticas são definidas pelos administradores com alto nível de abstração, ou seja, de forma independente das características específicas das tecnologias usadas nos elementos gerenciados. O módulo PEP deve ser implementado em cada elemento gerenciado. O módulo PDP interpreta as políticas e gera ações de controle que são aplicadas pelo PEP nos elementos correspondentes. Além disso, um módulo PEP pode enviar para o PDP notificações de eventos ou informações sobre a topologia da rede. Para a tomada de decisões de gerência, o PDP considera as políticas e também informações reportadas pelos módulos PEP. Para que as políticas e decisões de gerência possam ser tratadas em alto nível de abstração, o módulo PEP local deve ser responsável por lidar com as características particulares do elemento de rede quando da aplicação das ações de controle.

2.3 Gerência Autônômica

A abordagem PBM, reconhecidamente, promoveu avanços e foi usada com sucesso em diversas aplicações de gerência, principalmente gerência de segurança e de configuração, por exemplo, como apresentado em [13] e [47]. No entanto, essa abordagem apresenta algumas limitações entre as quais pode-se destacar a dificuldade na representação de políticas e na resolução de conflitos entre políticas [8, 12, 17]. Além disso, apesar de prover aos elementos de rede uma certa capacidade de reagir a alterações nas condições da rede, as políticas possuem uma natureza estática, uma vez que representam conjuntos de regras previamente definidas.

Diante de todas as limitações das abordagens mencionadas, foi proposto, recentemente, o conceito de Redes Autônômicas [10], que segue os princípios do paradigma de Computação Autônômica [37]. Em linhas gerais, pode-se definir um sistema autônômico como aquele capaz de gerenciar a si mesmo. O objetivo principal é que o sistema seja capaz de se adaptar, de forma autônoma, de acordo com o ambiente de operação. Neste contexto, as redes autônômicas são redes capazes de se auto-gerenciar, ou seja, redes com mecanismos de auto-gerência que controlam o comportamento da rede e seus serviços a fim de otimizar o desempenho do sistema com um todo.

A capacidade de adaptação dinâmica em sistemas autônômicos deve resultar de processos de aprendizagem e decisão, o que representa um avanço em relação à reconfiguração baseada em políticas. No entanto, essas abordagens não são conflitantes entre si. São, na verdade, complementares. Como discutido em [32], a evolução de sistemas baseados na arquitetura PBM são vistos como uma solução promissora para a implementação de redes autônômicas. De fato, redes autônômicas representam um paradigma que define propriedades e requisitos que podem ser implementados de diversas formas, utilizando-se para isso de diversas abordagens de gerência.

2.3.1 Propriedades de Auto-Gerência

Uma rede autônômica dispensa o papel de uma entidade gerenciadora centralizada, como ilustrado na Figura 2.3. Como um sistema autônômico, a rede é formada por um conjunto de elementos autônômicos. O termo elemento autônômico é genérico e não se restringe a elemento de rede. Ele pode ser qualquer componente da infraestrutura de rede, incluindo equipamentos e serviços, como uma impressora ou um servidor *Web*, por exemplo.

O paradigma de computação autônômica define quatro propriedades de auto-gerência para que um elemento seja considerado autônômico. Essas propriedades são denominadas funções de auto-gerência ou serviços autônômicos. Um *gerente autônômico* implementado em cada elemento provê a esses elementos as funções de auto-gerência. Assim, os elementos são capazes de gerenciar seu próprio estado e comportamento automaticamente sem a

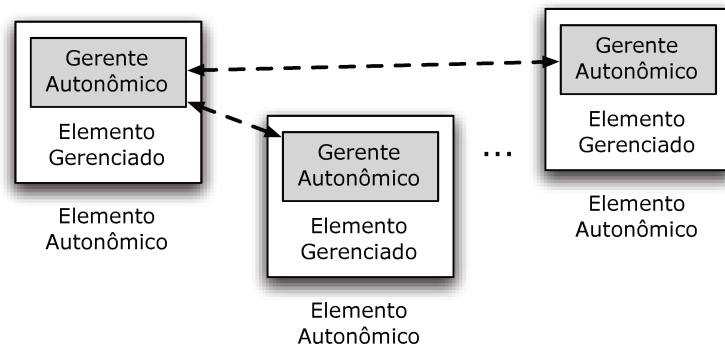


Figura 2.3: Rede formada por elementos autônomos.

intervenção de um administrador de rede. A seguir, é apresentada uma descrição sucinta das principais funções básicas de auto-gerência, como descrito em [37]:

- *Auto-configuração*: refere-se à capacidade do elemento de adaptar dinamicamente sua configuração de acordo com objetivos de alto-nível. Esses objetivos são definidos pelos administradores, por exemplo, a partir de regras de negócio ou políticas administrativas da organização.
- *Auto-otimização*: permite ao elemento ajustar continuamente seus parâmetros de controle com a finalidade de aumentar o desempenho do sistema como um todo. O elemento deve ser capaz de identificar e aprender formas e oportunidades de melhorar o desempenho em termos de eficiência ou custo.
- *Auto-cura*: refere-se à capacidade do elemento de identificar, diagnosticar e tratar eventuais falhas.
- *Auto-proteção*: é a capacidade do elemento de lidar com ações maliciosas, ataques e intrusões, ou mesmo, com efeitos em cascata resultantes de falhas. O ideal é que o elemento seja capaz de agir de forma pro-ativa antecipando as ações contra esses problemas.

Além dessas funções básicas de auto-gerência, várias outras foram propostas de acordo com requisitos e contextos específicos de diferentes sistemas autônomos. Em [10], é apresentada uma relação extensa de outras importantes funções de auto-gerência como: auto-consciência, auto-aprendizado, auto-diagnóstico e auto-organização.

2.3.2 Laço de Controle

O gerente autônômico implementa os processos de aprendizagem e decisão através de um laço de controle. Na Figura 2.4, é ilustrada a arquitetura clássica para um laço de controle [37]. O gerente autônômico consiste em quatro módulos funcionais e uma base de informações. Os módulos de monitoramento e análise são responsáveis pelo processo de aprendizagem, que permitem ao elemento construir a base de informações de gerência denominada Base de Conhecimento (*Knowledge Base*). Os módulos de planejamento e execução são responsáveis pela tomada de decisões de gerência e aplicação das ações correspondentes.

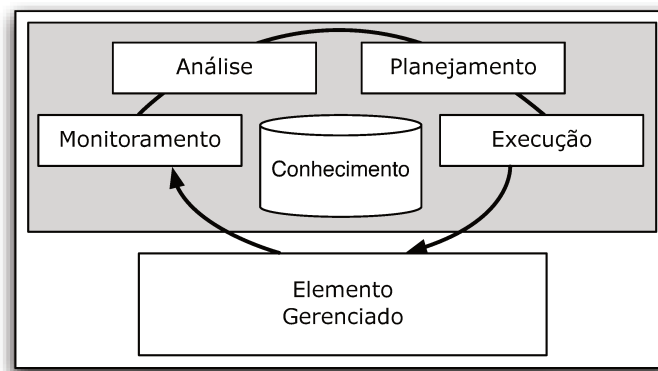


Figura 2.4: Arquitetura de um elemento autônomo.

O módulo de *monitoramento* é responsável por reconhecer o estado do elemento (seus parâmetros de configuração) e seu contexto de operação. Podem ser usados mecanismos sofisticados de reconhecimento de contexto ou mesmo coleta de informações do ambiente. No módulo de *análise*, essas informações são processadas para construção da base de conhecimento e para diagnóstico de falhas e anomalias. As decisões de gerência são realizadas pelo módulo de *planejamento*. De acordo com o elemento específico onde o gerente autônômico está implementado, as decisões podem resultar em uma ampla variedade de ações de controle como reconfiguração de parâmetros, atualização de *software*, ou bloqueio de porta. Este módulo também é responsável por definir ordem e prioridade das ações. Por fim, o módulo de *execução* é responsável pela efetiva aplicação das decisões tomadas. Este módulo deve lidar com as tecnologias e características específicas do elemento gerenciado para executar as ações da forma mais adequada.

O laço de controle é local e não depende de uma entidade centralizada. Obviamente, a base de conhecimento pode ser incrementada pelos administradores de rede. Para isso,

um administrador pode utilizar políticas de alto-nível de forma semelhante à abordagem PBM, mas, no caso de um sistema autônomo, a figura do administrador não existe em tempo de operação. Uma vez implementado o sistema, os elementos são capazes de se adaptar de forma automatizada e distribuída.

O projeto de uma rede autônoma pode envolver outras funções de auto-gerência além das quatro funções básicas mencionadas anteriormente. Algumas funções podem implementar propriedades emergentes resultantes da interação coordenada entre elementos autônomos. Nestes casos, o gerente autônomo deve incluir um módulo de comunicação, para permitir sinalização e compartilhamento de informações entre os elementos autônomos conforme necessário.

2.3.3 Desafios em Redes Autônomas

Apesar dos progressos obtidos, ainda são necessários avanços significativos para que de fato sejam desenvolvidas soluções de gerência distribuídas e sofisticadas para lidar com a complexidade das redes atuais. O paradigma de Computação Autônoma definiu requisitos e propriedades para sistemas auto-gerenciáveis, mas por ora não existem soluções definitivas para que funções de auto-gerência possam ser implementadas e exploradas em todo seu potencial. As abordagens para o projeto de funções de auto-gerência descritas na literatura não atendem completamente os requisitos de sistema autônomos [20, 31, 48]. A seguir, são relacionadas algumas dessas limitações:

- A falta de técnicas de monitoramento e diagnóstico de contexto que permitam a previsão ou identificação antecipada de anomalias e assim, viabilizem soluções de gerência proativa, ao invés de mecanismos capazes apenas de reagir a mudanças depois que o desempenho do sistema já foi comprometido;
- A dificuldade para representação do conhecimento (da semântica de informações relevantes para as tarefas de gerência) [9].
- A falta de mecanismos descentralizados que habilitem os elementos de rede com capacidade de adaptação;
- A falta de protocolos de comunicação e coordenação entre elementos autônomos (para viabilizar, por exemplo, funções de auto-organização em redes sem fio);
- A existência de mecanismos que não podem ser aplicados de forma genérica pois foram projetados para arquiteturas ou protocolos específicos.
- A existência de mecanismos demasiado complexos para serem implementados em equipamentos com recursos limitados;

- A complexidade de se descrever modelos analíticos que demonstrem a eficiência de mecanismos de controle descentralizados [36].

2.4 Considerações Finais

Diante deste breve histórico, pode-se observar que as abordagens de gerência têm evoluído no sentido de superar duas limitações principais: a falta de escalabilidade e a dependência de intervenção humana. As arquiteturas têm se tornado cada vez mais descentralizadas na tentativa de melhorar o grau de escalabilidade das soluções de gerência. Esse aspecto é evidente quando se analisa, em particular, o processo de tomada de decisão. Na abordagem tradicional, o sistema de gerência é usado para monitoramento e depois para aplicação das ações nos elementos de rede, mas as decisões sobre quais ações devem ser aplicadas ainda são tomadas pelo administrador. Na abordagem PBM, o processo de decisão foi, em certo grau, automatizado pelo PDP. O conceito de redes autonômicas avança ainda mais deslocando o processo de decisão para o próprio elemento de rede. As funções de auto-gerência nos elementos autonômicos permitiriam a implementação de processos avançados, como gerência proativa.

Grande parte dos problemas de desempenho nas redes atuais são conseqüência das limitações dos protocolos TCP/IP que foram projetados há muitos anos para cenários de rede que não contemplavam a dinâmica e a heterogeneidade das redes atuais. A maioria das limitações decorre da incapacidade de adaptação dos protocolos. Assim, é de extrema importância o desenvolvimento de redes autonômicas, com mecanismos que viabilizem a reconfiguração dinâmica de elementos de rede de forma escalável. Redes convergentes, com serviços sofisticados, obviamente, dependem dessa capacidade dos dispositivos de continuamente reagir às mudanças nas condições da infraestrutura de comunicação que afetam o desempenho do sistema.

O presente trabalho representa uma tentativa de se avançar o conhecimento sobre o projeto de mecanismos distribuídos de aprendizagem e decisão que permitam o desenvolvimento de soluções de auto-configuração. Tais soluções podem melhorar o desempenho de serviços de rede mesmo diante da heterogeneidade e dinâmica das infraestruturas de comunicação atuais.

Capítulo 3

Trabalhos Relacionados

Várias abordagens para mecanismos de otimização de desempenho [23] e gerência autônômica [44] foram propostas pela comunidade de pesquisa na área de redes de comunicação. Atualmente, existem várias iniciativas de pesquisa com foco no projeto de soluções com suporte a mecanismos de auto-gerência. Um exemplo de destaque foi o projeto ANA (Autonomic Network Architecture) [6], que envolveu diversas instituições de pesquisa da União Européia. Este projeto resultou no desenvolvimento de uma nova arquitetura de rede autônômica que contempla abstrações e primitivas de comunicação genéricas para o suporte a adaptação dinâmica da rede.

O foco do projeto ANA e de outras importantes propostas de arquitetura de redes autônômicas [48] é a arquitetura da rede em si para suporte a funções de auto-gerência. Este trabalho, em particular, trata de como realizar mecanismos de auto-configuração. Dessa forma, a presente proposta será comparada com outros trabalhos que apresentam soluções para reconfiguração dinâmica de protocolos.

3.1 Abordagens de Auto-Configuração

Em [50], Sutton *et al.* apresentaram uma arquitetura para suporte a reconhecimento de contexto e adaptação dos elementos de rede. Esta arquitetura define camadas de rede e recursos dos elementos como componentes em diferentes níveis. Um módulo denominado “gerente de componentes” coordena o monitoramento desses componentes e, através de um processo cognitivo, efetua a adaptação da pilha de protocolos, de acordo com o estado da rede.

Demestichas *et al.* [18] propuseram uma plataforma, denominada m@ANGEL, baseada nos princípios de computação autônômica para prover conectividade contínua em redes de acesso sem fio heterogêneas. Nesta proposta, os autores descrevem requisitos no nível de negócios e a arquitetura que provê funções de gerência autônômica tais como:

monitoramento de contexto, representação de perfis e contratos de serviços de rede e estratégias para negociação e configuração dos elementos de rede.

Outra arquitetura voltada para redes sem fio foi proposta por Bouet *et al.* [7]. Os autores introduziram uma arquitetura cognitiva que usa algoritmos de inteligência artificial para prover reconfiguração dinâmica em redes de acesso sem fio heterogêneas. Mais especificamente, essa arquitetura distribuída, denominada Cognitive Network Manager (CNM), utiliza Lógica Fuzzy para implementar mecanismos de decisão que reconfiguram os elementos de rede a partir de informações de monitoramento da rede. Uma das principais contribuições do trabalho é uma avaliação experimental da arquitetura em uma rede de acesso heterogênea composta por estações base WiMAX e pontos de acesso WiFi. Neste experimento, a arquitetura CNM foi aplicada na alocação dinâmica de canais e no balanceamento de carga através de *handover* vertical.

Em [41], o autor discute as limitações inerentes da arquitetura TCP/IP e propõe a mudança para um novo paradigma de rede cognitiva denominado “rede inteligente programável por software”(SPIN – *software programmable intelligent network*). Esta solução inclui abordagens utilizadas em diferentes arquiteturas de rede, como a rede de telefonia pública comutada, a Internet e também redes de telefonia celular. A arquitetura utiliza canais dedicados para tráfego de controle e elementos de software que são capazes de reconfigurar dinamicamente os elementos de rede em resposta às mudanças nas condições da rede.

A arquitetura Cognitive Complete Knowledge Network (CogNet) foi proposta por Manoj *et al.* em [43]. Essa solução considera o uso de algoritmos de decisão, inferência e mineração de dados, baseados em técnicas estatísticas, para extrair informações de grandes volumes de dados de observações da rede. Essa arquitetura segue uma abordagem *cross-layer* e os dados de monitoramento do estado da rede podem ser obtidos localmente ou a partir de um repositório de informações do sistema que armazena as observações dos elementos de rede. O conhecimento gerado a partir da análise dessas informações permite ao sistema CogNet usar algoritmos de tomada de decisão para otimizar parâmetros de configuração dos protocolos e prover o uso eficiente dos recursos da rede.

No trabalho apresentado em [32], Jennings *et al.* propuseram uma arquitetura para a gerência autônoma de redes. Esta arquitetura, denominada FOCALÉ, faz uso de modelos de informação de gerência e ontologias para prover ao sistema capacidade de decisão e aprendizagem sobre si mesmo e o seu ambiente de operação. O conhecimento do sistema, representado em modelos de alto nível de abstração, é utilizado por sistemas de gerência baseada em políticas para reconfigurar dinamicamente os elementos de rede em resposta a mudanças na rede. Esse processo de reconfiguração é realizado por meio de um laço de controle autônomo no qual o sistema monitora o contexto de operação e aplica ações de gerência conforme necessário.

3.2 Comparação Qualitativa

A seguir, é apresentada uma comparação entre a abordagem proposta nesta tese e os trabalhos relacionados em termos das propriedades dessas abordagens. Na Tabela 3.1, apresenta-se um resumo das principais características de cada arquitetura.

Abordagem	Camadas Alvo	Representação do Conhecimento	<i>Cross-Layer</i>	Descen-tralizada	Compatível com TCP/IP	Suporte a Requi-sitos de Aplicação
Sutton [50]	Todas	Simbólica	✓			
m@ANGEL [18]	Inferiores	Numérica		✓		
CNM [7]	Inferiores	Simbólica	✓	✓	✓	
SPIN [41]	Todas	Simbólica		✓		
CogNet [43]	Todas	Simbólica	✓		✓	
FOCALE [32]	Todas	Simbólica	✓		✓	✓
CogProt	Todas	Numérica	✓	✓	✓	✓

Tabela 3.1: Comparação entre arquiteturas de redes cognitivas.

Primeiramente, são identificadas as camadas da pilha de protocolos onde as abordagens podem ser aplicadas. A maioria das soluções podem ser aplicadas na reconfiguração de parâmetros em qualquer camada. As exceções são as arquiteturas m@Angel e CNM, direcionadas para as camadas de enlace e física de redes sem fio. Em seguida, caracteriza-se a natureza do formato de representação do conhecimento. Neste caso, considera-se duas classes: representação simbólica e numérica. A primeira classe refere-se ao uso de modelos de informações com alto nível de abstração e suporte à semântica dos dados, como políticas ou ontologias. A representação numérica significa que o conhecimento é representado por informações estatísticas de acordo com métricas ou domínios bem definidos. Por exemplo, métricas de desempenho de rede, como *jitter* e vazão agregada, ou mesmo métricas no nível de aplicação, como métricas de desempenho para fluxo (*streaming*) de vídeo.

Além disso, analisou-se quais arquiteturas suportam reconfiguração *cross-layer* e quais são descentralizadas. Algumas arquiteturas, como a CogNet, são parcialmente centralizadas, no sentido em que o processo de reconfiguração é descentralizado mas eles dependem de informações de repositórios centralizados. Foram consideradas como descentralizadas, somente as abordagens cujo processo de ajuste dos protocolos pode ser completamente realizado de forma descentralizada. Outra propriedade relevante é a compatibilidade com as tecnologias de rede atuais de ampla utilização. Neste contexto, foi verificado quais abordagens são compatíveis com as redes TCP/IP. São consideradas compatíveis as soluções que não requerem alterações na arquitetura TCP/IP ou na operação de seus protocolos.

Por fim, foi verificado se o projeto das arquiteturas estudadas permite que o processo de reconfiguração dinâmica possa levar em conta requisitos e contexto de aplicações. Esse aspecto é muito importante, por exemplo, no contexto de redes de acesso sem fio, onde o ajuste dinâmico dos protocolos implementados nos dispositivos móveis pode melhorar significativamente o desempenho de uma aplicação.

A estratégia fundamental sobre a qual todas as abordagens foram desenvolvidas consiste em monitorar o estado da rede e, então, decidir sobre as ações de gerência apropriadas para reconfigurar os elementos de rede. Desse modo, as arquiteturas propostas apresentam muitas características em comum. Elas descrevem módulos funcionais e requisitos para o projeto de redes com mecanismos de auto-gerência. No entanto, existem duas limitações importantes nas abordagens descritas na literatura. Em primeiro lugar, algumas dessas abordagens são baseadas em uma arquitetura centralizada que depende de uma visão geral da rede. Embora essas soluções possam de fato determinar valores ótimos para parâmetros de configuração dos protocolos, elas resultam em sistemas que envolvem algoritmos e processos muito complexos (que exigem alto poder computacional e manipulação de grandes volumes de dados). Consequentemente, essas soluções possuem baixo grau de escalabilidade e são tão complexas quanto os sistemas de gerência atuais oferecidos pela indústria dos fabricantes de equipamentos de rede. Em segundo, a maioria das soluções considera processos de aprendizagem e tomada de decisão para reconfigurar os elementos, mas elas não descrevem uma técnica viável para implementação de tais processos. Algumas propostas indicam estratégias que poderiam ser utilizadas para isso, principalmente, técnicas na área de inteligência artificial. No entanto, não é apresentada uma alternativa para decidir sobre como reconfigurar dinamicamente os protocolos de acordo com as informações obtidas do monitoramento da rede. Uma exceção é a arquitetura CNM que foi implementada e testada em um ambiente de rede real, porém, essa solução foi projetada para cenários de aplicação específicos em redes de acesso sem fio.

A principal contribuição desta tese é uma clara descrição de como implementar um processo cognitivo de uma maneira viável e efetiva para prover aos elementos de rede a capacidade de tomar decisões de reconfiguração apropriadas. A presente proposta consiste em um arcabouço descentralizado para o suporte a mecanismos de auto-configuração com o objetivo de melhorar o desempenho médio do sistema. O processo de ajuste dinâmico dos parâmetros pode ser direcionado por requisitos da aplicação, se conveniente. Esta solução pode ser aplicada a uma grande variedade de parâmetros de configuração de protocolos em diferentes camadas. Ela pode ser implantada nas redes atuais, uma vez que não são necessárias alterações nos protocolos em uso. Além disso, essa implantação pode ser feita de forma incremental, pois elementos de rede que implementam a solução e os que não implementam podem interagir entre si.

Capítulo 4

CogProt

Sob certa perspectiva, seria ideal não haver necessidade da realização de tarefas de administração de rede. Isto é, poderia ser possível projetar sistemas de computação e comunicação capazes de se auto-organizar e de se adaptar dinamicamente de forma ótima sem qualquer intervenção humana. Essa é uma das perspectivas implícita nos princípios do paradigma de Computação Autônoma [30], inspirado na fisiologia do corpo humano. Seres humanos não precisam se preocupar com funções vitais durante o cotidiano. Tais funções, como a respiração e a circulação sanguínea, são controladas pelo Sistema Nervoso Autônomo. No entanto, modelos computacionais inspirados na natureza não são novidade. O que é inovador naquele paradigma é a visão holística da “autonomicidade” do sistema como um todo; ao contrário de um foco na mecânica da autonomia e sua construção através de algoritmos.

Quando o sistema autônomo em questão é uma rede de dados, temos o que se denomina uma Rede Autônoma. Para que uma rede seja autônoma ela deve suportar funções de auto-gerência. Assim, ela seria capaz de gerenciar a si mesma sem a intervenção humana. Os serviços e protocolos de uma rede autônoma devem ser capazes de se adaptar às condições da infra-estrutura de comunicação subjacente. Prover propriedades de auto-gerência que emergem da interação coordenada entre elementos autônomos é uma tarefa muito complexa. O projeto de sistemas autônomos ainda é um grande desafio para a comunidade científica, dada a complexidade de se modelar mecanismos distribuídos de reconhecimento de contexto, aprendizagem e decisão.

Neste capítulo, introduz-se uma abordagem cognitiva para a implementação de mecanismos de auto-configuração. A solução proposta consiste em um arcabouço, denominado CogProt, para o projeto de mecanismos de auto-configuração descentralizados. O objetivo de tais mecanismos é habilitar os elementos de rede com capacidade de reconfiguração dinâmica, para que eles possam se adaptar às mudanças na rede evitando, assim, degradação no desempenho dos serviços de comunicação.

4.1 Visão Geral da Abordagem Cognitiva

Desde a definição de propriedades de auto-gestão no contexto do paradigma de Computação Autônoma, várias soluções foram propostas para o projeto de mecanismos de gestão avançados com suporte a reconfiguração de elementos de rede. Algumas soluções foram baseadas em abordagens oriundas de disciplinas como Teoria do Controle [19, 29] e Teoria dos Jogos [27, 33, 42]. Embora avanços importantes tenham sido alcançados, a implementação de mecanismos descentralizados de auto-gestão não é uma realidade e ainda demanda esforços de pesquisa.

Mais recentemente, mecanismos cognitivos têm sido utilizados com sucesso no ajuste dinâmico de protocolos de rede. O conceito de Redes Cognitivas é uma extensão do conceito de Rádios Cognitivos [26], cujo foco é o ajuste de parâmetros nas camadas física e de enlace em redes sem fio para prover compartilhamento eficiente do espectro de frequências disponível. Redes Cognitivas [23, 52] envolvem o uso de algoritmos cognitivos e abordagens *cross-layer* para prover otimização dinâmica do sistema de comunicação como um todo [38]. Algoritmos cognitivos incluem mecanismos sofisticados de aprendizagem e decisão para otimizar o desempenho de sistemas de comunicação complexos.

Neste contexto, este trabalho introduz uma abordagem cognitiva para a reconfiguração dinâmica de protocolos. Em termos concretos, esta abordagem consiste em um arcabouço, denominado CogProt, que provê aos elementos de rede funções de auto-configuração. Na Figura 4.1, são ilustradas algumas características da solução proposta como explicado a seguir.

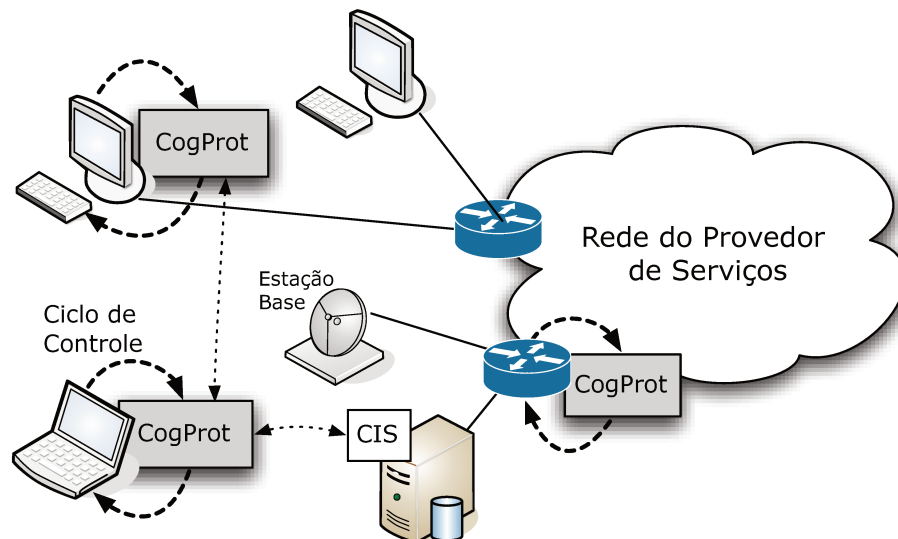


Figura 4.1: Visão geral da abordagem cognitiva.

O CogProt pode ser implementado nos mais diversos elementos de rede, como roteadores ou sistemas finais, móveis ou não. Elementos habilitados com o Cogprot, então denominados *elementos de rede cognitivos*, podem operar juntamente com elementos que não o implementam, uma vez que o funcionamento dos protocolos não é alterado. O processo de auto-configuração consiste em um ciclo de controle que implementa as tarefas de aprendizagem e decisão. O CogProt monitora periodicamente o desempenho e realiza ações de reconfiguração para reagir às variações nas condições da rede.

Além do ajuste dinâmico, o CogProt pode ser utilizado na configuração inicial dos protocolos. Por exemplo, quando um elemento se conecta a uma rede local, ele pode obter informações de configuração de elementos cognitivos vizinhos ou de um Serviço de Informações Cognitivas (*Cognitive Information Service* – CIS). Tal serviço é considerado nesta abordagem como uma alternativa de compartilhamento de informações de gerência entre os elementos cognitivos quando conveniente (conforme discutido em mais detalhes na Seção 4.5).

4.2 O Processo de Auto-Configuração

O arcabouço CogProt é implementado localmente em cada elemento de rede. Seus componentes são apresentados na Figura 4.2. O processo de auto-configuração consiste em um ciclo de controle que provê a reconfiguração periódica dos protocolos de acordo com um histórico recente de desempenho. Este processo caracteriza-se por uma abordagem *cross-layer* [49], uma vez que ele pode monitorar e reconfigurar parâmetros de várias camadas. Informações de uma camada podem ser utilizadas para decisão sobre reconfiguração de protocolos em outra camada.

O processo de auto-configuração é denominado cognitivo porque envolve tarefas de aprendizagem e decisão. O processo de aprendizagem consiste no monitoramento do desempenho e construção de uma Base de Conhecimento durante a operação da rede. Esse conhecimento reflete o histórico recente de desempenho e é utilizado no processo de tomada de decisão, que controla a reconfiguração dos parâmetros de interesse. Além das informações locais na Base de Conhecimento, um elemento cognitivo pode, opcionalmente, usar informações remotas no processo de auto-configuração. Essas informações podem ser compartilhadas diretamente entre elementos cognitivos vizinhos ou indiretamente, através de um serviço CIS eventualmente disponível na rede local.

Como ilustrado na Figura 4.3, o ciclo de controle do processo de auto-configuração consiste em três etapas: *análise*, *decisão* e *ação*. Esse processo permite o ajuste dinâmico de parâmetros de configuração dos protocolos de rede. Um parâmetro de interesse é periodicamente reconfigurado em função do desempenho observado.

Na etapa de ação, um valor aleatório é atribuído ao parâmetro de interesse. Para

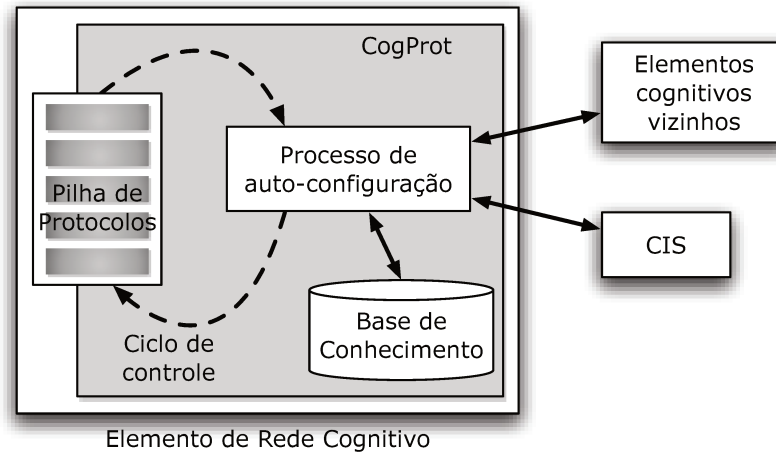


Figura 4.2: Arquitetura de um elemento de rede cognitivo.

isso, utiliza-se um gerador aleatório com distribuição de probabilidade normal. Na etapa de análise, são coletadas informações de desempenho associadas ao parâmetro de acordo com uma *métrica de desempenho* pré-definida. As informações de desempenho coletadas a cada iteração são usadas para atualizar a Base de Conhecimento, de forma que essa base reflita um histórico recente de desempenho. Na etapa de decisão, a base de conhecimento é analisada para que se possa identificar, entre os valores já usados na configuração do parâmetro, aquele que apresenta o melhor desempenho médio. O melhor valor é então definido como a média do gerador de números aleatórios, de forma que esse valor seja atribuído, com maior frequência, ao parâmetro de interesse (na etapa de ação).

Após um período de seguidas iterações, a base de conhecimento permite identificar o valor que apresenta o melhor desempenho nas condições correntes. Com o uso de valores aleatórios na fase de ação, o elemento cognitivo pode perceber mudanças nas condições da rede, pois essas mudanças terão impacto no desempenho observado na base de conhecimento para cada valor do parâmetro. Uma questão crítica é como implementar as tarefas de aprendizagem e decisão de forma escalável e descentralizada. A seguir, são explicadas como estas tarefas são realizadas na presente proposta, de acordo com as etapas do processo de auto-configuração.

4.2.1 Análise

O CogProt periodicamente ajusta o parâmetro de interesse de acordo com o conhecimento adquirido em oscilações aleatórias na configuração desse parâmetro. A cada iteração do ciclo de controle um valor aleatório é escolhido para o parâmetro de interesse. O

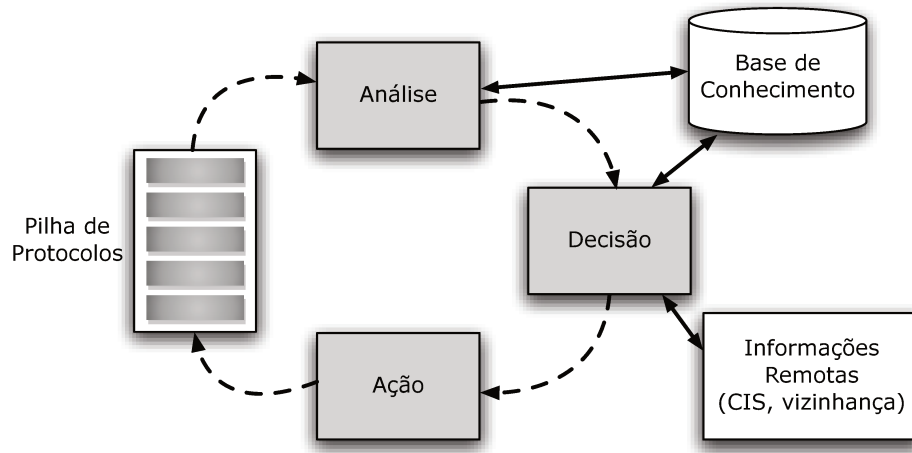


Figura 4.3: Etapas do processo de auto-configuração.

CogProt monitora o desempenho obtido com a configuração corrente e armazena esta informação na base de conhecimento. As informações acumuladas na base são usadas na definição de como se ajustar apropriadamente o parâmetro. O processo cognitivo de aprendizagem consiste na construção (contínua) dessa base de conhecimento. Essa tarefa é realizada na etapa de análise do ciclo de controle. Nesta etapa, são coletadas informações de desempenho associadas ao parâmetro de interesse de acordo com uma métrica de desempenho pré-definida.

A etapa de análise é ilustrada na Figura 4.4. Considere o ajuste dinâmico de um parâmetro de interesse p , que pode ser um parâmetro de configuração qualquer como a potência do sinal de transmissão em uma interface de rede sem fio ou o tamanho inicial da janela de congestionamento do TCP. Para simplificar a explicação, considera-se o domínio de p como um conjunto enumerável. Neste caso, considere que p pode assumir um valor $p_i \in \{p_1, \dots, p_n\}$. O procedimento pode ser adaptado para os casos onde p pode assumir um valor contínuo (por exemplo, um valor real em um intervalo $[p_{min}, p_{max}]$).

O primeiro passo na etapa de análise é medir o desempenho obtido com o valor corrente de p . Este valor é um valor aleatório atribuído ao parâmetro na etapa de ação na iteração anterior do ciclo. Seja τ o intervalo de repetição do ciclo. O desempenho é medido após esse intervalo desde a atribuição do valor na etapa anterior. Para se medir o desempenho, deve ser definida uma função d que calcula o desempenho de acordo com uma métrica específica. O desempenho obtido com o valor corrente do parâmetro p é denotado por $d(p)$. A métrica deve ser escolhida de acordo com a aplicação específica e o parâmetro de interesse. No caso de protocolos da camada de aplicação ou transporte, é importante que a métrica permita verificar o desempenho fim-a-fim. Por exemplo, a métrica pode ser o

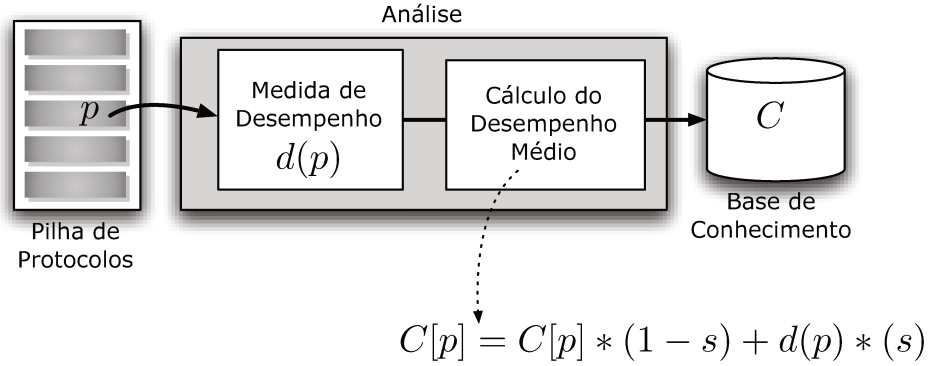


Figura 4.4: Monitoramento de desempenho na etapa de análise.

atraso médio em um fluxo ou taxa de perda de quadros na camada de enlace.

As informações de desempenho são armazenadas na base de conhecimento C . Essa base mantém o desempenho médio para cada valor possível de p . Assim, $C[p_i]$ é o desempenho médio para cada valor p_i entre os valores que podem ser atribuídos a p . Para calcular o desempenho médio é utilizada uma média ponderada exponencial (*exponentially weighted moving average* – EWMA). A cada iteração do ciclo, o desempenho médio do valor corrente do parâmetro, p_c , é atualizado de acordo com a Fórmula 4.1, onde s , $0 < s \leq 1$, é um peso (*smoothing factor*) que determina a relevância da última medida de desempenho na média geral.

$$C[p_c] = C[p_c] \times (1 - s) + d(p_c) \times (s) \quad (4.1)$$

Inicialmente, a base pode ser inicializada com valores nulos. Durante a operação da rede, valores aleatórios são atribuídos a p . A cada iteração do ciclo, o desempenho médio do valor corrente do parâmetro é atualizado. Assim, a Base de Conhecimento C reflete o histórico de desempenho para os valores possíveis de p . Neste caso, a base C cresce linearmente com o tamanho do conjunto $\{p_1, \dots, p_n\}$. No caso de um parâmetro que pode assumir valores reais, é necessário adaptar essa estratégia. A solução mais direta é segmentar a faixa de valores possíveis para p e calcular o desempenho médio por segmento.

O peso s pode ser usado para controlar o efeito da medida de desempenho mais recente. Valores altos de s aumentam o efeito do valor mais recente na média geral. Isso permite ao mecanismo reagir às mudanças nas condições da rede de forma mais rápida. Valores baixos de s resultam em um comportamento conservador que pode conferir mais estabilidade ao sistema, porque, neste caso, o impacto de estados transitórios é amenizado.

4.2.2 Decisão

Na Figura 4.5, são apresentados os detalhes da etapa de decisão. O primeiro passo nesta etapa é determinar o melhor valor (p^*) para o parâmetro de interesse. Isso é feito através de uma busca na base de conhecimento pelo valor p_i que apresenta o melhor desempenho. Além da base de conhecimento local, informações remotas podem ser consideradas na escolha desse valor. Possíveis fontes dessas informações são elementos cognitivos vizinhos ou um Serviço de Informações Cognitivas disponível na rede local (CIS – *Cognitive Information Service*). Por exemplo, no momento em que um dispositivo sem fio se conecta a uma rede, a base de conhecimento ainda não contém informações suficientes para a tomada de decisão. Neste caso, o melhor valor pode ser determinado pelo conhecimento já acumulado por outros nós na rede.

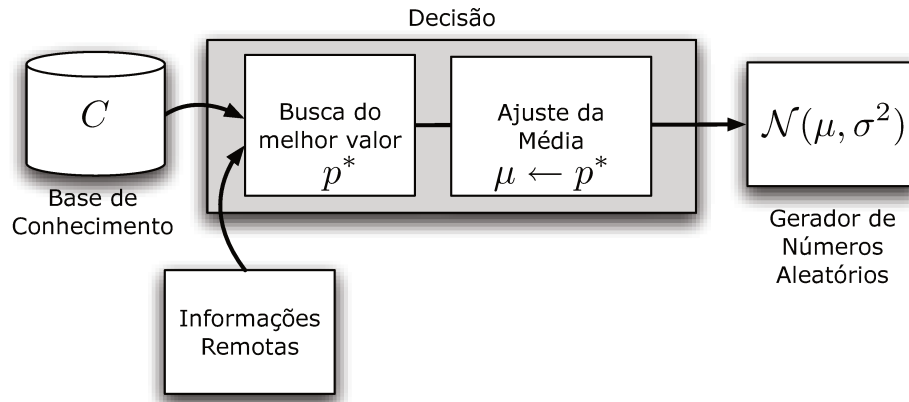


Figura 4.5: Ajuste da média μ na etapa de decisão.

O valor p^* determinado no passo anterior é então definido como a média (μ) de um gerador de números aleatórios com uma distribuição de probabilidade normal. Na realidade, o objetivo aqui é manter a média da distribuição normal como o melhor valor conhecido para p . Assim, o processo cognitivo de decisão consiste em atualizar a média quando necessário. Caso um novo valor seja identificado como o melhor valor de p , esse valor deve ser atribuído à média. É importante observar que o parâmetro p não é reconfigurado nesta etapa. Apenas a média da distribuição normal é ajustada a fim de que ela assumo o melhor valor conhecido para p .

O processo de decisão não depende de uma entidade centralizada. A determinação do melhor valor é feita localmente. O uso de informações remotas é uma alternativa que pode ser importante em alguns cenários. Essa característica contribui para um maior grau de escalabilidade do mecanismo.

4.2.3 Ação

O parâmetro p somente recebe um valor na etapa de ação. Nesta etapa, um novo valor aleatório é obtido do gerador de números aleatórios. Este valor é, então, atribuído a p . Estes passos são ilustrados na Figura 4.6.

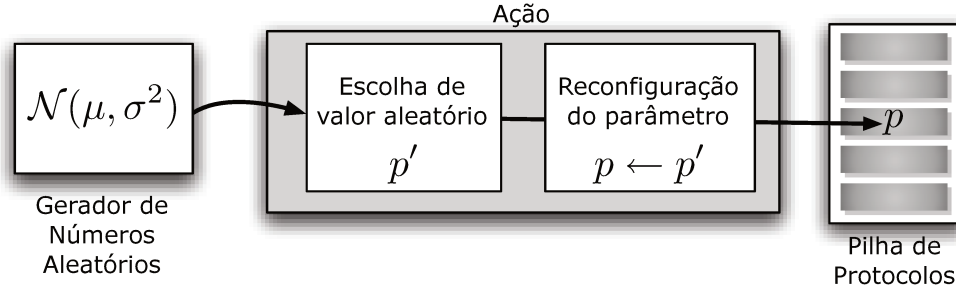


Figura 4.6: Ajuste do parâmetro na etapa de ação.

Um valor aleatório p' é gerado de acordo com uma distribuição normal, com média μ e desvio padrão σ . A média dessa distribuição foi ajustada na etapa anterior para ser igual ao melhor valor conhecido p^* . De acordo com a função densidade de probabilidade da distribuição normal padrão, sabe-se que aproximadamente 68% das ocorrências de uma variável aleatória estão no intervalo $[\mu - \sigma, \mu + \sigma]$ e que aproximadamente 99% estão no intervalo $[\mu - 3\sigma, \mu + 3\sigma]$, como ilustrado na Figura 4.7. Assim, p receberá, com maior probabilidade, um valor igual ou próximo do melhor valor conhecido. Por outro lado, as ocorrências de p' distantes do melhor valor permitem ao mecanismo reagir às mudanças. Quando acontecem mudanças que levam a um novo valor ótimo, esse valor será eventualmente utilizado e seu desempenho ótimo terá um reflexo na base de conhecimento, o que por sua vez resultará no ajuste da média na fase de decisão.

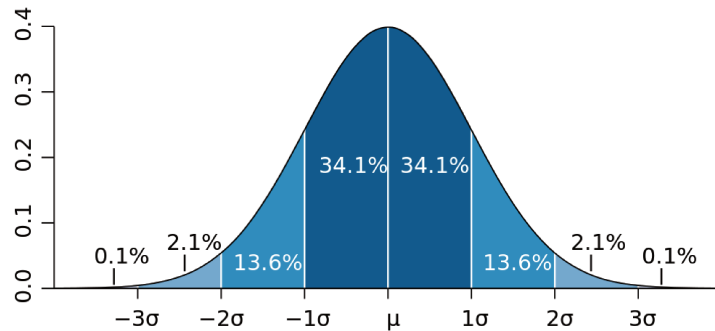


Figura 4.7: Função densidade para distribuição normal padrão.

O desvio padrão (σ) pode ser usado para controlar a agressividade do mecanismo. Quanto menor o desvio padrão, menos agressivo é o mecanismo em testar novos valores para p . Isso significa, que levará mais tempo para que seja possível perceber mudanças na rede. Um desvio padrão alto pode diminuir o tempo de reação do mecanismo, mas, por outro lado, pode gerar instabilidade. Portanto, o desvio padrão afeta diretamente o tempo de convergência e a estabilidade do mecanismo.

4.3 Descrição do Processo como Algoritmo

A abordagem proposta pode ser usada no projeto de mecanismos de reconfiguração de diversos parâmetros de configuração de protocolos de rede. Para projetar um mecanismo para um parâmetro de configuração específico, é necessário caracterizar esse parâmetro de interesse e também alguns parâmetros do próprio ciclo de controle do CogProt.

Os elementos que definem o parâmetro de interesse são apresentados na Tabela 4.1. O valor padrão é usado na primeira iteração do ciclo para inicializar o parâmetro p e a média da distribuição normal. A métrica m determina como deve ser medido o desempenho na etapa de análise.

Tabela 4.1: Definição do parâmetro de interesse.

p	Parâmetro que será reconfigurado dinamicamente.
$p_{default}$	O valor padrão para p .
$\{p_1, \dots, p_n\}$	Domínio de p (valores que p pode assumir).
m	Métrica de desempenho associada a p .

Os parâmetros de controle do CogProt são relacionados na Tabela 4.2. Não existe um valor padrão para essas variáveis. Elas devem ser definidas no contexto de cada aplicação do CogProt de acordo com o parâmetro de interesse e a métrica de desempenho correspondente.

Tabela 4.2: Parâmetros de controle do processo de auto-configuração.

τ	Intervalo de repetição do ciclo de controle.
s	Peso associado ao desempenho medido para o valor corrente de p .
σ	O desvio padrão para a função de distribuição normal.

O intervalo τ está diretamente relacionado com o monitoramento de desempenho na fase de análise. Por exemplo, se a métrica de desempenho é a vazão média, a vazão será calculada em relação ao intervalo τ a cada repetição do ciclo. Como discutido na

subseção anterior, os valores de s e σ influenciam diretamente o tempo de convergência e a estabilidade do mecanismo.

A seguir, é apresentado um algoritmo genérico para implementação do processo de auto-configuração. Inicialmente, o parâmetro p e a média μ recebem o valor padrão. A partir daí, o processo de reconfiguração repete as etapas de análise, decisão e ação a cada intervalo τ . Este intervalo deve ser determinado de acordo com os requisitos da aplicação e o nível da operação de reconfiguração (por exemplo, no nível de fluxo ou pacotes).

Algorithm 1 Processo de auto-configuração de um parâmetro p

Require: $p_{default} \in [p_1, \dots, p_n], \tau, \sigma, s \in (0, 1]$

Ensure: Reconfiguração dinâmica de p

```

1:  $p \leftarrow p_{default}$ 
2:  $\mu \leftarrow p_{default}$ 
3: repeat
4:    $wait(\tau)$ 
5:    $x \leftarrow d(p)$ 
6:    $C[p] \leftarrow C[p] \times (1 - s) + x \times s$ 
7:    $\mu \leftarrow p_i | \max\{C[p_i], i = 1, \dots, n\}$ 
8:    $p \leftarrow normal(\mu, \sigma)$ 
9: until forever

```

À medida que as iterações ocorrem, a base de conhecimento C acumula o desempenho médio para cada valor possível de p . Neste caso, $C[p_i]$ é o desempenho médio de $p_i \in \{p_1, \dots, p_n\}$. Na etapa de análise (linhas 5 e 6), calcula-se o desempenho obtido com o valor corrente de p e atualiza-se a base C . A função para mensurar o desempenho ($d()$) deve ser implementada para cada aplicação de acordo com a métrica m pré-definida. Na etapa de decisão (linha 7), a média μ é ajustada de acordo com o melhor valor de p conhecido, ou seja, o valor p_i com maior desempenho médio na base de conhecimento. Por fim, p é reconfigurado com um valor aleatório na etapa de ação (linha 8).

O objetivo deste algoritmo não é determinar o valor ótimo de p . Esta solução é apropriada para os casos onde não é possível ou é inviável (computacionalmente) determinar um valor ótimo global para um parâmetro de configuração. O que este processo de auto-configuração faz é adaptar o parâmetro durante a operação da rede com o objetivo de aumentar o desempenho médio do sistema. O algoritmo ajusta continuamente a média do gerador aleatório com o valor que provê o melhor desempenho de acordo com a Base de Conhecimento. Consequentemente, o melhor valor conhecido para as condições atuais do sistema será escolhido com maior frequência na etapa de ação.

Mudanças no estado da rede são percebidas porque diferentes valores de p são eventualmente utilizados. Se um valor qualquer p_i passa a prover o melhor desempenho em um novo cenário, esse desempenho será observado e resultará na atualização da média da

distribuição normal. Consequentemente, esse valor será escolhido com maior frequência. Dessa forma, as oscilações aleatórias permitem que a própria média seja ajustada apropriadamente. Como uma regra geral, o mecanismo busca manter a média da distribuição normal como o melhor valor conhecido para o parâmetro de interesse. A média é continuamente deslocada para que o melhor valor conhecido seja escolhido com maior frequência.

O tempo que o mecanismo demora para perceber mudanças na rede depende do valor dos parâmetros do ciclo de controle. Por exemplo, um desvio padrão alto aumenta a probabilidade de que um valor p_i distante da média seja testado, o que diminui o tempo de percepção de uma eventual mudança nas condições da rede. No entanto, este caso também diminui a probabilidade com que o melhor valor é escolhido para ser atribuído a p , o que afeta negativamente o desempenho médio. Existe, portanto, um compromisso (*trade-off*) entre tempo de convergência e estabilidade, em termos do desvio padrão.

4.4 Exemplo

A seguir, é apresentado um exemplo numérico para ilustrar o funcionamento do ciclo de controle do CogProt. Seja p um parâmetro de configuração de algum protocolo. Considere o domínio de p o conjunto $\{1, 2, 3, 4, 5\}$ e $p_{default} = 2$. Na Figura 4.8, são ilustradas 7 iterações do ciclo de controle. Na figura, são representados o valor corrente de p , a média da distribuição normal μ e a base de conhecimento C , a cada iteração do ciclo. Neste exemplo, são usados valores arbitrários para o desempenho medido na etapa de análise ($d()$), assim como para o valor aleatório gerado na etapa de ação. Uma vez que os dados obtidos nestas etapas foram escolhidos arbitrariamente (para fins de ilustração), os valores de σ e τ são irrelevantes. Para cálculo do desempenho médio, considera-se $s = 0.8$.

Inicialmente, o valor padrão é atribuído a p e μ . A base de conhecimento é inicializada com valores nulos para o desempenho médio de cada valor possível para p . Após um intervalo τ , temos a primeira iteração no instante t_1 , ilustrado na Figura 4.8(b). Na etapa de análise, atualiza-se o desempenho médio para o valor corrente do parâmetro (neste caso, $p = 2$). Assim, a base é atualizada conforme o cálculo a seguir:

$$\begin{aligned} C[2] &= C[2] * (1 - s) + d(2) * (s) \\ C[2] &= 0.0 * 0.2 + 75.0 * 0.8 \\ C[2] &= 60.0 \end{aligned}$$

Na etapa de decisão, o mecanismo verifica se é preciso atualizar μ , o que não é necessário neste momento pois a média já é igual ao melhor valor de p conhecido. Por fim, um valor aleatório é escolhido para o parâmetro de interesse a partir da distribuição normal. Esses passos se repetem a cada iteração. Em particular, no instante t_6 , é ilustrado o

caso no qual o mecanismo decide atualizar a média, pois a base de conhecimento permite identificar um novo valor com melhor desempenho médio. Esse novo valor ($p = 4$) foi testado na interação anterior e resultou em um melhor desempenho. O número de interações necessárias para que o mecanismo encontre o melhor valor, depende de s e σ . Neste caso, por exemplo, o desempenho médio do valor 4 não superaria o do valor padrão se s fosse igual a 0.6 (mais iterações seriam necessárias). Por outro lado, o valor 4 poderia ter sido testado em uma interação anterior, se o desvio padrão fosse mais alto.

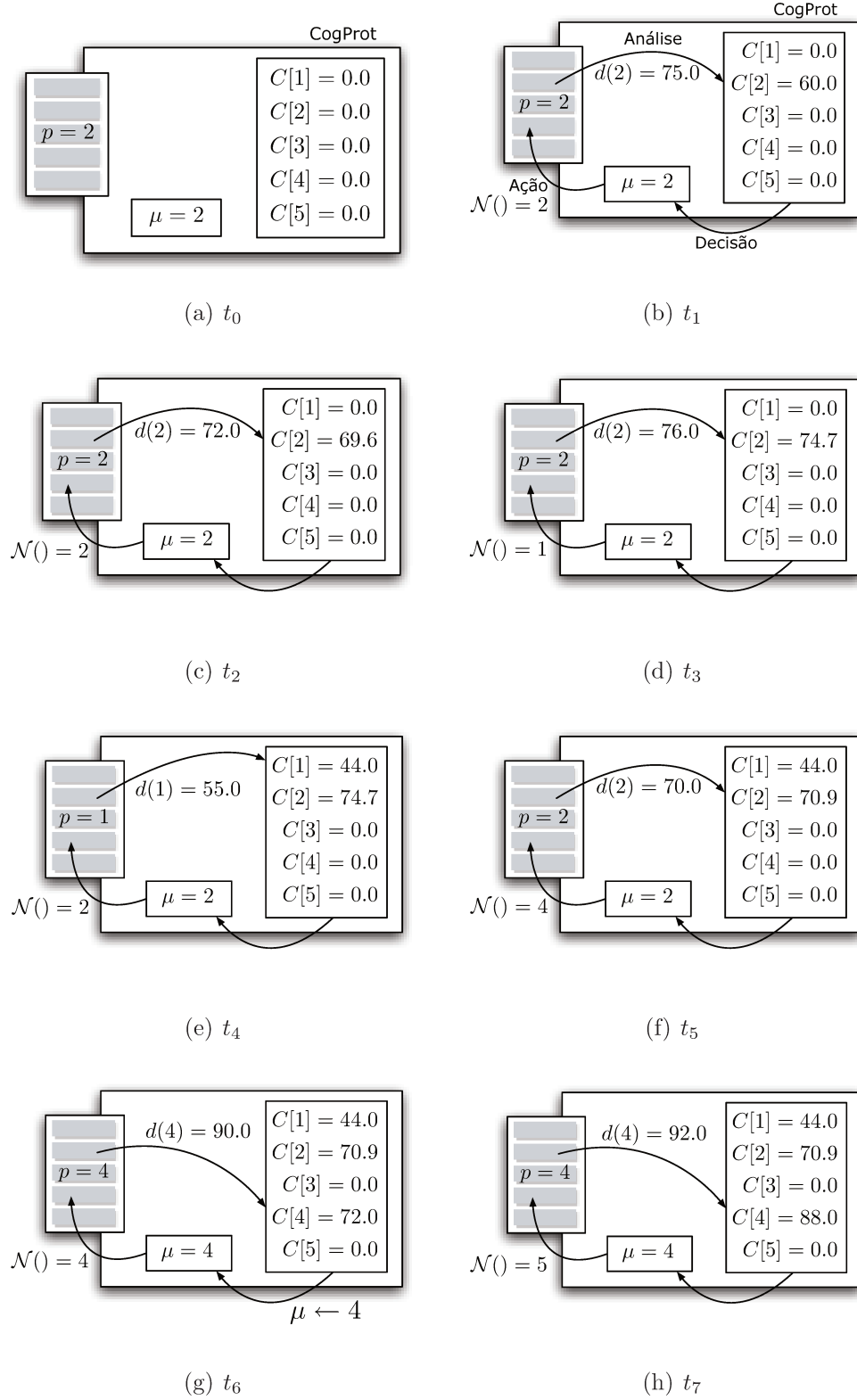


Figura 4.8: Exemplo numérico.

4.5 Cenários de Aplicação

A pilha de protocolos TCP/IP inclui uma grande variedade de protocolos cujos parâmetros de configuração precisam ser apropriadamente ajustados, especialmente em cenários heterogêneos e dinâmicos. A abordagem proposta pode ser usada na reconfiguração dinâmica de diversos parâmetros. Para cada aplicação específica da abordagem proposta, deve ser escolhido um parâmetro de interesse que se deseja ajustar e a métrica de desempenho correspondente.

Na Tabela 4.3, são apresentados alguns exemplos de parâmetros de protocolos amplamente usados na redes atuais e possíveis métricas de desempenho que podem ser associadas a esses parâmetros. Como discutido na Seção 4.3, a abordagem proposta é mais adequada na reconfiguração de parâmetros para os quais a seguinte propriedade é satisfeita. Considere que, em um dado contexto, p_t é um valor ótimo para p , $p \in \{p_0, p_1, \dots, p_n\}$. Neste caso, para $u > t$, $d(p_u) \geq d(p_{u+1})$, e para $s < t$, $d(p_s) \geq d(p_{s-1})$, ou seja, valores próximos do ótimo apresentam melhor desempenho em relação a valores mais distantes do ótimo. Isso permite que a percepção de mudanças seja eficiente mesmo com um desvio padrão pequeno. Um desvio padrão pequeno é importante para que o melhor valor conhecido seja usado com mais frequência, o que aumenta o desempenho médio.

Tabela 4.3: Exemplos de parâmetros para aplicação do CogProt.

Camada	Parâmetro	Métrica de Desempenho
Aplicação	<i>video streaming bitrate</i>	<i>jitter</i>
Transporte	janela de congestionamento do TCP	vazão, atraso, <i>jitter</i>
	fator de incremento da janela do TCP	
	<i>TCP start threshold</i>	
Rede	<i>OLSR willingness</i>	taxa de perda de pacotes
Enlace	<i>retry Limit</i>	vazão, SNR, <i>frame error rate</i>
	<i>RTS/CTS threshold</i>	
Física	potência de transmissão (WiFi)	

A métrica de desempenho pode ser definida de acordo com os requisitos de QoS ou QoE das aplicações. Além disso, um mecanismo de auto-configuração pode exigir múltiplas métricas de desempenho. O mecanismo pode alternar entre as métricas de acordo com os requisitos da aplicação. Por exemplo, uma instância do CogProt usada para auto-configuração de um protocolo de transporte poderia usar duas métricas de desempenho, uma para aplicações de dados e outra para aplicações de vídeo ou voz. Além disso, a métrica de desempenho não precisa ser limitada a uma métrica simples. Ela pode ser definida por uma estrutura mais sofisticada como, por exemplo, uma métrica composta

utilizando múltiplos critérios ou objetivos [35], ou mesmo uma função utilidade que de alguma forma possa refletir a justiça no compartilhamento de recursos [46].

A escolha apropriada da métrica de desempenho é importante para que o mecanismo seja capaz de melhorar o desempenho médio dos serviços de rede. É também muito importante formular uma métrica que possa ser calculada apenas com informações locais para que se possa garantir um maior grau de escalabilidade e desempenho do mecanismo de auto-configuração correspondente.

Além da definição da métrica de desempenho, cada aplicação do CogProt exige a definição de valores apropriados para os parâmetros do ciclo de controle (especificados na Tabela 4.2). Por exemplo, o intervalo de iteração do ciclo de controle (τ) para um parâmetro na camada de aplicação poderia ser da ordem de segundos (no nível de fluxos), enquanto para um parâmetro na camada de enlace esse intervalo poderia ser da ordem de milissegundos (no nível de quadros).

O arcabouço CogProt pode ser aplicado na auto-configuração de múltiplos parâmetros simultaneamente. Como ilustrado na Figura 4.9, esses parâmetros podem estar na mesma camada (*intra-layer*) ou em camadas diferentes (*inter-layer*). Este segundo caso caracteriza uma solução *cross-layer*. Conforme a taxonomia explicada em [49], este caso se enquadra na categoria de arquitetura *cross-layer* com base de dados compartilhada. As informações de várias camadas são compartilhadas por meio da base de conhecimento. Por exemplo, informações sobre desempenho em termos de *frame error rate* na camada de enlace podem ser usadas no ajuste de parâmetros de configuração na camada de transporte.

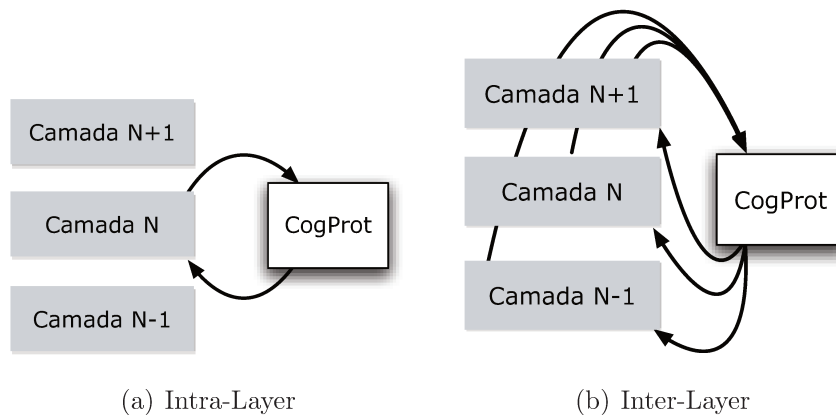


Figura 4.9: Cenários de auto-configuração de múltiplos parâmetros.

Como discutido até aqui, o processo de auto-configuração é intrinsecamente descentralizado, uma vez que ele é implementado localmente em cada elemento de rede. Entretanto, a disponibilidade de informações remotas pode aprimorar o processo de decisão

em aplicações específicas. Assim, um mecanismo de reconfiguração baseado no CogProt pode ser classificado como *local*, se os processos de aprendizagem e decisão dependem apenas da base de conhecimento local, ou *distribuído*, se informações adicionais obtidas remotamente são usadas nesses processos.

No caso de mecanismos distribuídos, o elemento de rede cognitivo pode obter informações adicionais de elementos vizinhos ou de um serviço CIS. Nesses casos, um aspecto importante a ser considerado é a abordagem utilizada para obtenção de informações remotas. Em princípio, podem ser consideradas duas arquiteturas: centralizada e descentralizada. No primeiro caso, ilustrado na Figura 4.10, faz-se necessário o serviço de informações cognitivas (CIS) e, conseqüentemente, um protocolo específico para permitir a comunicação entre o elemento cognitivo e o serviço CIS. Assim, além das informações de desempenho locais, o elemento pode obter do servidor políticas de configuração e valores padrão para parâmetros de interesse. Além disso, o serviço CIS pode mediar o compartilhamento de dados da base de conhecimento entre os elementos. Assim, um nó poderia conhecer os dados da base de conhecimento de um nó vizinho.

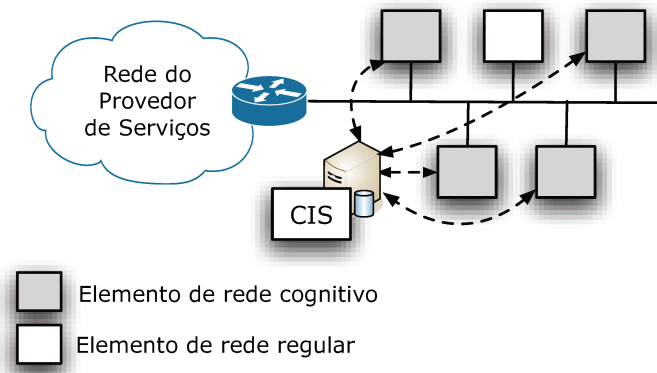


Figura 4.10: Arquitetura centralizada (baseada em servidor).

Essa abordagem é mais simples, do ponto de vista de implementação, mas, como qualquer arquitetura centralizada, inviabiliza um sistema em larga escala. No entanto, essa estratégia pode ser interessante em muitos cenários. Esse é o caso, por exemplo, de redes de acesso sem fio, com estações base que podem prover serviços de coordenação em favor dos nós móveis. Neste cenário, os elementos de rede podem usar o serviço CIS para compartilhar informações sobre a rede local, a fim de facilitar o processo cognitivo de auto-configuração. Adicionalmente, o serviço CIS pode implementar políticas de gerência para a configuração inicial dos nós, de forma similar a um serviço DHCP que provê configuração dinâmica para endereçamento IP em uma rede local. Por exemplo, quando

um nó se conecta à rede, ele pode requisitar ao serviço CIS um valor inicial apropriado para a média (μ) da distribuição normal. Esse valor pode ser determinado pelo serviço em função do histórico observado pelos demais nós na rede, ou seja, o CIS pode construir sua própria base de conhecimento a partir do desempenho monitorado por todos os nós associados a ele.

A outra alternativa é uma arquitetura distribuída, como exemplificado na Figura 4.11. Essa estratégia confere escalabilidade ao sistema, mas é mais complexa de ser implementada. Neste caso, é necessário definir um protocolo de comunicação para que os elementos de rede cognitivos possam compartilhar informações de gerência entre si. Esse protocolo deve incluir mecanismos para descoberta de elementos cognitivos e compartilhamento de informações de base de conhecimento.

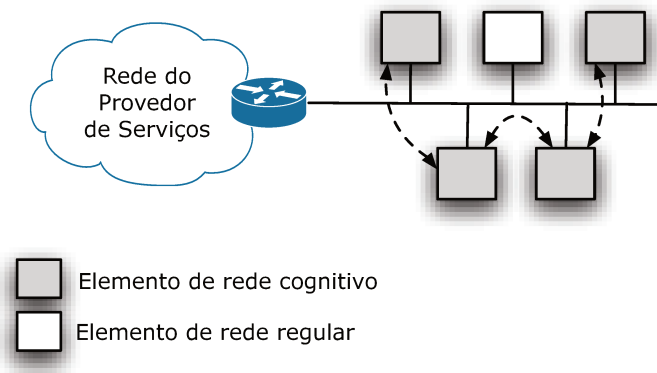


Figura 4.11: Arquitetura descentralizada.

Uma vez que é inviável atualizar elementos e serviços de rede em larga escala, é desejável que novos mecanismos tenham compatibilidade com os protocolos e serviços em operação. Assim, a coexistência entre elementos de rede cognitivos e os elementos tradicionais é uma questão chave para melhorar o desempenho do sistema de comunicação em uma perspectiva de longo prazo. A abordagem proposta é consistente com o princípios de redes TCP/IP de se restringir mecanismos sofisticados à borda da rede, mantendo os elementos no núcleo o mais simples quanto possível. O arcabouço CogProt adiciona apenas um pequeno grau de complexidade ao elemento de rede. Assim, ele pode ser implementado mesmo em elementos de rede com recursos limitados. Além disso, não são necessárias alterações no funcionamento dos protocolos, portanto, os elementos cognitivos podem interagir com elementos tradicionais e, dessa forma, a solução proposta pode ser implementada em uma rede de forma incremental.

Capítulo 5

Estudos de Caso

No Capítulo 3, introduziu-se uma avaliação qualitativa da abordagem proposta. Esta avaliação foi feita a partir de uma análise comparativa entre as características e funcionalidades do arcabouço CogProt e outras soluções de auto-configuração. Neste capítulo, apresenta-se uma avaliação quantitativa com o objetivo de verificar a eficácia do CogProt na reconfiguração dinâmica de protocolos de rede. Esta avaliação consiste em uma série de estudos de caso nos quais a abordagem proposta foi aplicada na auto-configuração de protocolos amplamente utilizados nas redes atuais: o protocolo TCP e o protocolo de acesso ao meio em redes sem fio 802.11.

Foram realizados quatro estudos de caso. No primeiro estudo, o arcabouço CogProt foi aplicado no controle da janela de congestionamento do TCP, como descrito na Seção 5.1. A abordagem cognitiva foi usada na reconfiguração dinâmica do fator de incremento da janela do TCP. Resultados de simulações mostram que a solução baseada no CogProt, denominada CogTCP, permite melhorar a vazão média desse protocolo. Em um segundo estudo de caso, a abordagem cognitiva foi usada no ajuste dinâmico da taxa de transmissão em redes WiFi, como apresentado na Seção 5.2. Além disso, a abordagem proposta foi avaliada no controle simultâneo de múltiplos parâmetros. Avaliou-se, em um mesmo cenário, a eficácia do CogProt no controle de ambos os parâmetros: o fator de incremento da janela do TCP e a taxa de transmissão na camada MAC. Os resultados de simulações são apresentados na Seção 5.3. Todas as simulações foram realizadas utilizando-se o Network Simulator (ns-2).

Além das simulações, foi realizado um experimento em um ambiente de rede real. O mecanismo CogTCP, avaliado através de simulações, foi implementado como um módulo do sistema Linux. Foram realizados experimentos com fluxos TCP em uma rede local e entre estações na UNICAMP e na Universidade Federal do Mato Grosso do Sul (UFMS). Os resultados são coerentes com as simulações, reafirmando o aumento da vazão média dos fluxos TCP (Seção 5.4).

5.1 Auto-Configuração do TCP

Na pilha de protocolos TCP/IP, o principal serviço da camada de transporte é a comunicação lógica entre dois processos. O TCP é um protocolo da camada de transporte orientado a conexão. Ele oferece dois serviços principais: transporte confiável de dados e controle de congestionamento. Em linhas gerais, existem duas classes de mecanismos de controle de congestionamento na camada de transporte. O *controle de congestionamento fim-a-fim* é aquele que não depende de um suporte da camada de rede. Por outro lado, existe o *controle de congestionamento assistido pela rede* que depende de informações explícitas sobre o estado da rede a partir de notificações da camada de rede.

O TCP implementa um mecanismo de controle de congestionamento fim-a-fim, cujo objetivo é limitar a taxa de transmissão da conexão de acordo com as condições da rede. Para isso, uma das variáveis de estado do TCP é a *janela de congestionamento* que define um limite para a quantidade de dados transmitidos mas cujo recebimento ainda não foi confirmado pelo destinatário através de uma mensagem de reconhecimento (ACK). O controle de congestionamento consiste em ajustar essa janela durante o tempo de vida da conexão de acordo com o grau de utilização da rede. Essa limitação na taxa de transmissão das conexões TCP tenta evitar a saturação da capacidade da rede e ao mesmo tempo compartilhar essa capacidade de forma justa entre as conexões.

O controle de congestionamento tradicional do TCP¹ consiste em três fases: partida lenta (*slow start*), prevenção de congestionamento (*congestion avoidance*) e recuperação rápida (*fast recovery*) [40]. Quando uma conexão TCP é iniciada, o tamanho da janela de congestionamento é igual a 1 MSS (*maximum segment size*) e a fase do controle de congestionamento é a de partida lenta. Nesta fase, a janela cresce exponencialmente até atingir um limiar específico (*slow start threshold*), quando então entra na fase de prevenção de congestionamento.

Na prevenção de congestionamento, o tamanho da janela é ajustado pelo mecanismo denominado algoritmo de aumento aditivo e diminuição multiplicativa (*additive increase and multiplicative decrease* – AIMD). A janela cresce linearmente de acordo com um fator de incremento (α), cujo valor padrão é igual a 1. Assim, o tamanho da janela é incrementado em 1 MSS a cada RTT (*round trip time*). Quando uma possível perda de segmento é identificada por recebimento de 3 ACKs duplicados, o mecanismo entra na fase de recuperação rápida. O tamanho da janela é reduzido de acordo com um fator de decremento multiplicativo igual a 2, ou seja, o tamanho da janela é reduzido à metade. Na fase de recuperação rápida, a janela é incrementada em 1 MSS quando um ACK duplicado é recebido. Se um novo ACK é recebido, o mecanismo volta para a fase de prevenção de congestionamento. Em todos os casos, quando uma perda de segmento é percebida por

¹Considera-se como tradicional o protocolo TCP Reno.

esgotamento de temporização, o mecanismo entra na fase de partida lenta com a janela ajustada para 1 MSS e o limiar ajustado para a metade da janela quando a perda foi identificada.

O objetivo do algoritmo de congestionamento é alcançar, sutilmente, o limite da capacidade de transmissão, sem saturar a banda disponível. Por isso, o crescimento é linear. No entanto, isso pode resultar em uma utilização ineficiente dos recursos, uma vez que a conexão pode levar muito tempo para usar toda a largura de banda disponível. Em alguns cenários, uma abordagem mais agressiva (com um fator de incremento maior) poderia resultar numa recuperação mais rápida do tamanho da janela após uma perda [1]. De fato, o controle da evolução da janela de congestionamento afeta diretamente a utilização da rede. Valores maiores de α são interessantes em redes com alto produto banda-atraso ou no caso em que o estado da rede é de níveis baixos ou moderados de utilização dos recursos. No entanto, não existe uma solução viável para se determinar, antecipadamente, a largura de banda disponível ou o nível de congestionamento em um caminho fim-a-fim em uma rede de larga escala como a Internet[16]. Consequentemente, é impraticável definir um valor (estático) para α que seja ótimo para qualquer cenário. Sendo assim, a reconfiguração dinâmica do fator de incremento da janela pode melhorar o desempenho da conexão. Essa foi a motivação deste estudo de caso.

Neste estudo, foi desenvolvido um mecanismo, denominado CogTCP, para ajuste dinâmico do fator de incremento da janela de congestionamento do TCP. O CogTCP é uma instância do arcabouço CogProt. O primeiro passo para aplicação da abordagem cognitiva é a definição do parâmetro de interesse, como descrito na Seção 4.3. Neste caso, o parâmetro de interesse p é o fator de incremento linear (α) da janela de congestionamento. O valor padrão é $p_{default} = 1$ e o domínio é $p \in \{1, \dots, 5\}$. A métrica de desempenho é a vazão média ².

Foram realizadas simulações para avaliar a eficácia do CogTCP em melhorar o desempenho médio do TCP³. O cenário de rede simulado é ilustrado na Figura 5.1. A topologia da rede consiste em um nó de origem S , dois roteadores R_1 e R_2 e dois nós de destino D_1 e D_2 . Para cada destino, os caminhos apresentam atraso de propagação e taxa de erro (PER – *Packet Error Rate*) diferentes. No enlace R_2 - D_1 , a taxa de erro é 5% e o atraso é 20ms. Já no enlace R_2 - D_2 , a taxa de erro é 8% e o atraso é 180ms. O mecanismo CogTCP foi habilitado no nó de origem S .

Foi comparado o desempenho do protocolo TCP regular usando os valores fixos de α com o desempenho do TCP reconfigurado dinamicamente através do CogTCP. Foram criados dois fluxos TCP: um fluxo entre os nós S e D_1 e outro entre os nós S e D_2 . Para

²Nesta tese, consideramos a vazão como a quantidade de dados efetivamente transmitidos com sucesso a um destinatário em um dado intervalo (essa métrica é também denominada *goodput*).

³A versão do TCP utilizada no simulador foi o TCP Newreno.

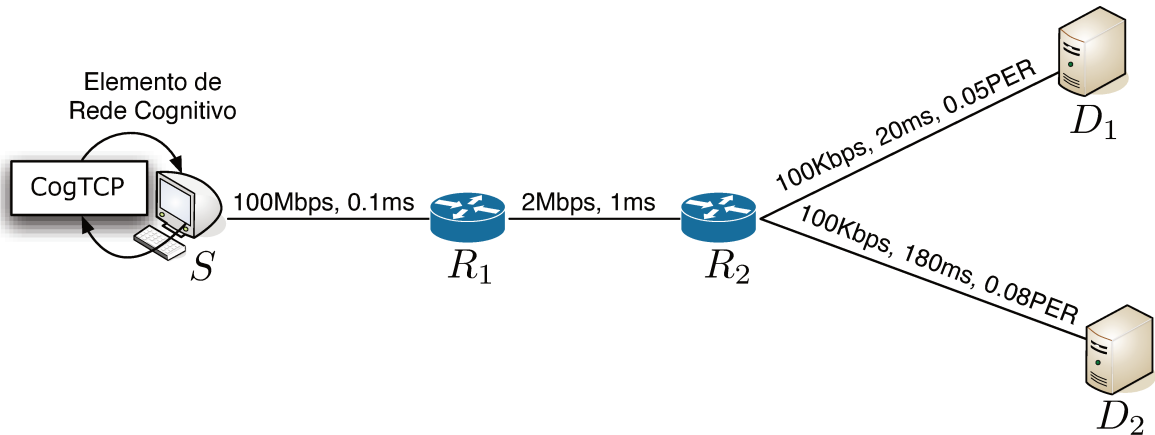


Figura 5.1: Cenário de rede simulado na auto-configuração do TCP.

os parâmetros de controle do próprio mecanismo, foram definidos os seguintes valores: $\tau = 1s$ (intervalo de iteração do ciclo de controle), $s = 0.5$ (*smoothing factor*) e $\sigma = 0.5$. Esses valores foram escolhidos empiricamente, a partir da realização de simulações com diferentes combinações de valores para esses parâmetros de controle. O tempo de duração da simulação é 1000s. Os resultados apresentados são uma média de várias iterações com sementes diferentes considerando um intervalo de confiança de 95%.

Foram considerados três cenários: apenas fluxo $S-D_1$, apenas fluxo $S-D_2$ e ambos os fluxos. Na Figura 5.2(a), são apresentados os resultados para o primeiro cenário. Pode-se notar que uma abordagem conservadora é mais eficiente para o fluxo com destino D_1 , ou seja, valores menores para α apresentam melhor desempenho. No entanto, para o fluxo com destino D_2 , valores maiores para α garantem melhor desempenho, como apresentado na Figura 5.2(b) para o segundo cenário. Neste caso, os valores maiores de α permitem uma recuperação mais rápida da janela diante de um RTT maior.

No terceiro cenário, considera-se que a origem S estabelece conexões com ambos os destinos. Na primeira metade da simulação, é estabelecido o fluxo $S-D_1$. Em seguida, durante a segunda metade da simulação, é estabelecido o fluxo $S-D_2$. Os resultados são apresentados na Figura 5.3. Neste cenário heterogêneo, o CogTCP oferece o melhor desempenho médio, pois qualquer valor fixo leva à degradação de desempenho em algum caso. A reconfiguração dinâmica de α garante o melhor desempenho na média, mesmo com apenas uma pequena diferença de desempenho entre os possíveis valores de α (observada nos dois primeiros cenários).

O comportamento do CogTCP é ilustrado na Figura 5.4. Essa figura mostra a vazão durante o tempo de execução de uma iteração das simulações para o cenário com ambos os fluxos. Durante a primeira metade da simulação, quando está ativo apenas o fluxo

$S-D_1$, pode-se observar que o $\alpha = 1$ leva ao melhor desempenho. No entanto, o desempenho desse valor não é o melhor na segunda metade da simulação, quando está ativo o segundo fluxo. De forma análoga, fixar $\alpha = 5$ leva ao mesmo problema. A reconfiguração dinâmica efetuada pelo CogTCP mantém o desempenho próximo do melhor durante toda a simulação, resultando, portanto no melhor desempenho médio.

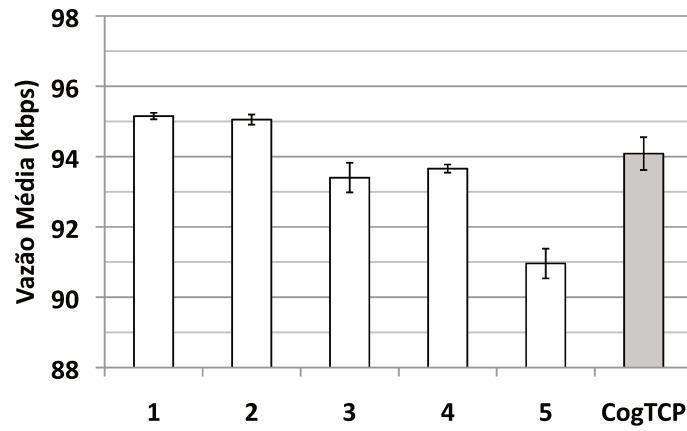
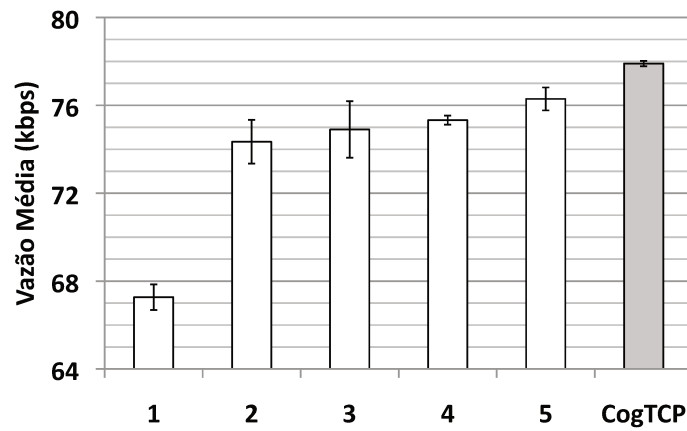
(a) Considerando apenas o fluxo $S-D_1$.(b) Considerando apenas o fluxo $S-D_2$.

Figura 5.2: Vazão média das conexões TCP para valores fixos de α e para o CogTCP.

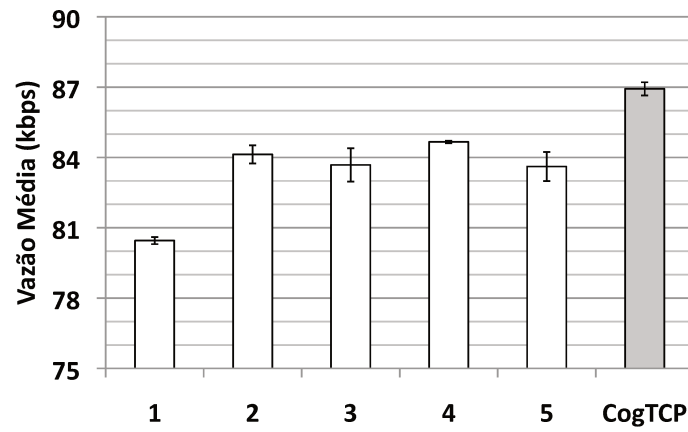


Figura 5.3: Vazão média agregada para valores fixos de α e para o CogTCP em um cenário heterogêneo.

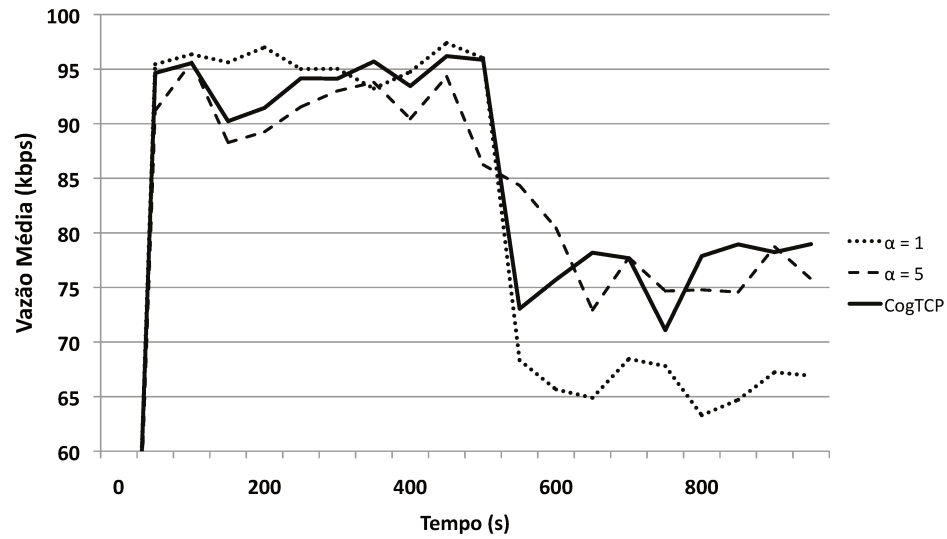


Figura 5.4: Vazão instantânea ao longo da simulação para CogTCP, $\alpha = 1$ e $\alpha = 2$ em um cenário heterogêneo.

5.2 Adaptação da Taxa de Transmissão

Entre os avanços importantes na área de comunicações sem fio, pode-se destacar o desenvolvimento de sofisticados esquemas de codificação e modulação do sinal de rádio. Uma combinação desses esquemas resulta em uma determinada capacidade nominal de transmissão de dados. Alguns esquemas de modulação são capazes de prover altas taxas de transmissão. No entanto, eles são mais sensíveis às flutuações nas condições do canal e não são eficientes para transmissões de longa distância.

Por outro lado, esquemas de modulação mais robustos permitem uma maior cobertura de comunicação, mas resultam em baixas taxas de transmissão. As interfaces de redes sem fio atuais suportam múltiplas taxas de transmissão. As constantes mudanças nas condições da rede exigem o ajuste dinâmico da taxa de transmissão. Isso motivou o desenvolvimento de algoritmos para controle da taxa de transmissão. O objetivo desses algoritmos é otimizar o desempenho da rede através da escolha adequada da taxa de transmissão que provê o melhor desempenho dadas as condições atuais do canal.

De acordo com a abordagem utilizada para estimar a condição do canal, os algoritmos de controle de taxa podem ser classificados como: *baseados em estatística* ou *baseados em qualidade do sinal*. Os primeiros estimam a qualidade do canal de acordo com informações estatísticas sobre a transmissão de dados, por exemplo, o número médio de retransmissões. Os algoritmos baseados na qualidade do sinal utilizam medições sobre o sinal transmitido no meio como indicadores das condições do canal. Neste caso, o parâmetro mais comum é a razão entre sinal e ruído (SNR – *signal-to-noise ratio*).

Um dos mais conhecidos algoritmos baseados em estatística é o *Auto Rate Fallback* (ARF), proposto por Kamerman e Monteban [34]. Este algoritmo incrementa a taxa de transmissão quando é alcançado um número específico de sucessos em tentativas consecutivas de transmissão de quadros. De forma análoga, a taxa é decrementada quando ocorre um certo número de perdas consecutivas na transmissão de quadros. Este algoritmo apresenta instabilidade, pois, mesmo depois de escolher a taxa ótima, ele continua tentando incrementar a taxa após alcançar o limiar de transmissões bem sucedidas.

No contexto dos algoritmos baseados na qualidade do sinal (SNR), pode-se destacar o algoritmo *Receiver-Based Auto Rate* (RBAR) proposto por Holland *et al.* [28]. Nessa solução, o nó receptor envia mensagens de controle ao nó emissor indicando a qualidade do sinal recebido. A partir dessas informações, o nó emissor escolhe a melhor taxa. Essa solução exige o uso do mecanismo RTS/CTS⁴ para troca das informações de controle que indicam a qualidade do sinal no receptor. Essa abordagem apresenta algumas limitações como a perda de desempenho devido à sobrecarga (*overhead*) introduzida pelo mecanismo

⁴O mecanismo RTS/CTS (*request to send/clear to send*) é usado em redes sem fio 802.11 para evitar colisões no acesso ao meio compartilhado.

RTS/CTS e pelas informações de controle adicionais, a necessidade de alterações no protocolo para incluir as informações sobre qualidade do sinal e a complexidade de se mensurar a qualidade do sinal e associar a mesma com as taxas disponíveis [2, 3].

Em geral, as limitações dos mecanismos de controle de taxa estão relacionadas com a técnica utilizada para avaliar as condições e estimar a qualidade do canal de transmissão. Estimativas imprecisas acabam por induzir erros ou um longo tempo de convergência no ajuste da taxa de transmissão. Os principais problemas são a dificuldade em diferenciar perdas (a fim de se evitar ajustes desnecessários na taxa) e o uso de métricas muito complexas, ou mesmo não realísticas, para estimar a qualidade do canal.

Neste estudo de caso, o arcabouço CogProt foi usado no projeto de um novo mecanismo para o controle da taxa de transmissão denominado CORA (*Cognitive Rate Adaptation*). O mecanismo CORA foi projetado para otimizar o desempenho da camada MAC a partir do ajuste dinâmico da taxa de transmissão de dados. Neste caso, o parâmetro de interesse é a taxa de transmissão. A métrica de desempenho foi definida como a vazão média na camada MAC. Os parâmetros de controle do CORA foram definidos empiricamente com os seguintes valores: $\tau = 100ms$, $s = 0,9$ e $\sigma = 0.3$.

O desempenho do mecanismo proposto foi comparado com os algoritmos ARF e RBAR. Em todas as simulações, foram utilizados o padrão IEEE 802.11g e o protocolo de roteamento pró-ativo *Destination-Sequenced Distance Vector* (DSDV). O tempo de simulação foi de 1200s. Os resultados apresentados são a média de pelo menos 10 iterações com um intervalo de confiança de 95%. No padrão 802.11g, existem 8 esquemas de modulação correspondentes às seguintes taxas nominais: 6, 9, 12, 18, 24, 36, 48 e 54Mbps.

No cenário simulado, ilustrado na Figura 5.5, a topologia consiste em dois nós: um nó estacionário E e um nó móvel M . O nó M move com velocidade constante de $0.25m/s$ afastando-se do nó E até alcançar uma distância de $300m$ entre eles. Durante toda a simulação, temos um fluxo *constant bit rate*(CBR) de $20Mbps$ do nó E para M usando o protocolo *User Datagram Protocol* (UDP).

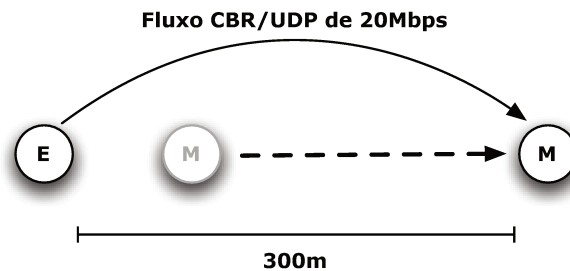


Figura 5.5: Cenário de rede simulado para avaliação do CORA.

Em primeiro lugar, foi avaliado o desempenho para cada uma das taxas disponíveis sem a utilização de nenhum mecanismo de ajuste dinâmico. Os resultados são apresentados na Figura 5.6. Com o aumento da distância, aumenta a atenuação do sinal e a taxa de perdas. Como esperado, as taxas mais baixas suportam transmissões a distâncias mais longas. Pode-se observar que para diferentes distâncias existe uma taxa que é melhor. Isso justifica que o controle da taxa tem uma importância crítica para se evitar degradação no desempenho. Um mecanismo de controle da taxa deve ser capaz de sustentar o desempenho máximo ao selecionar a melhor taxa de acordo com o cenário correspondente.

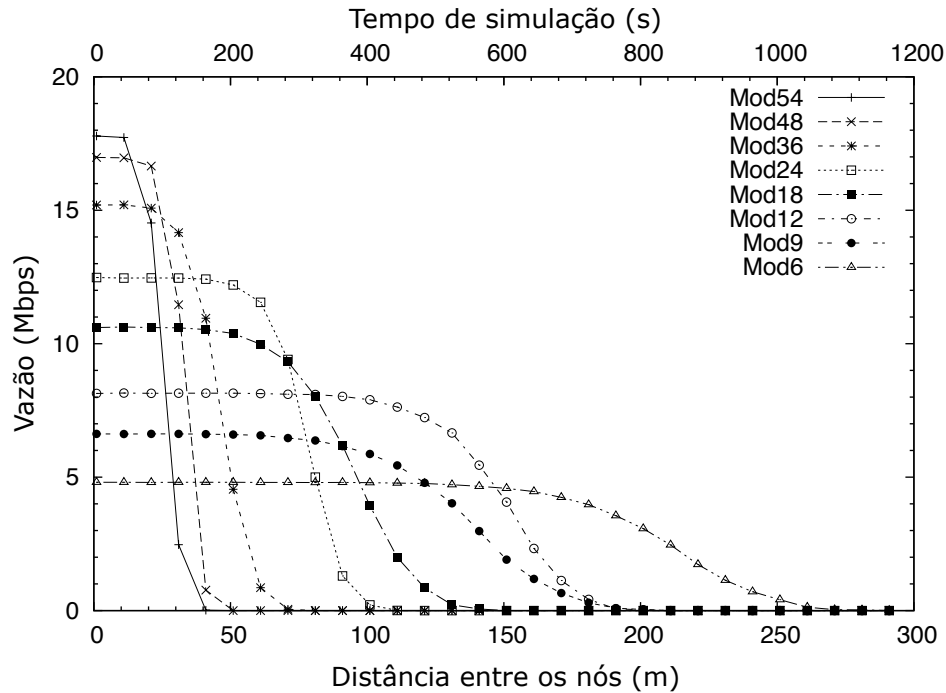


Figura 5.6: Vazão média para as taxas 802.11g.

Depois desse estudo preliminar, foi feita uma comparação entre os mecanismos de controle da taxa. Na Figura 5.7, apresenta-se a vazão de cada mecanismo em função do tempo/distância. Nessa figura, também mostram-se as curvas correspondentes ao desempenho para cada taxa apresentada no gráfico da Figura 5.6. O mecanismo proposto apresenta melhor desempenho que os concorrentes ao convergir de forma mais rápida para a taxa ótima a cada distância.

Para o mesmo experimento, foi também avaliada a vazão média, como apresentado na Figura 5.8. O gráfico mostra a vazão média obtida em cada algoritmo de controle e em cada taxa fixa. Neste caso, o CORA apresentou um desempenho 4,5% melhor que o ARF e 11,2% melhor que o RBAR.

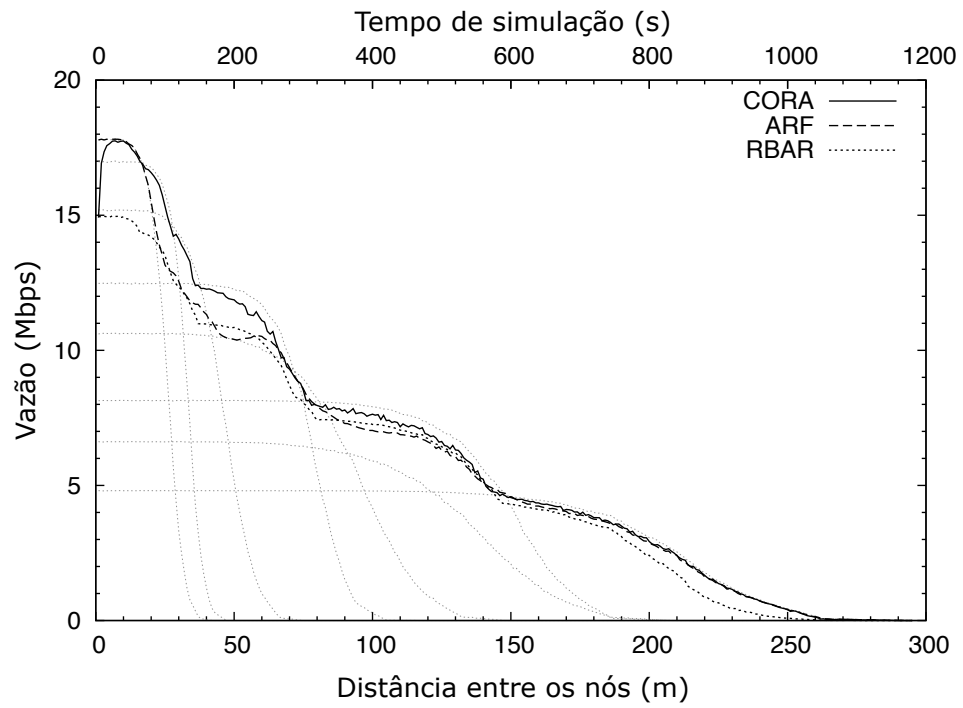


Figura 5.7: Vazão em função do tempo.

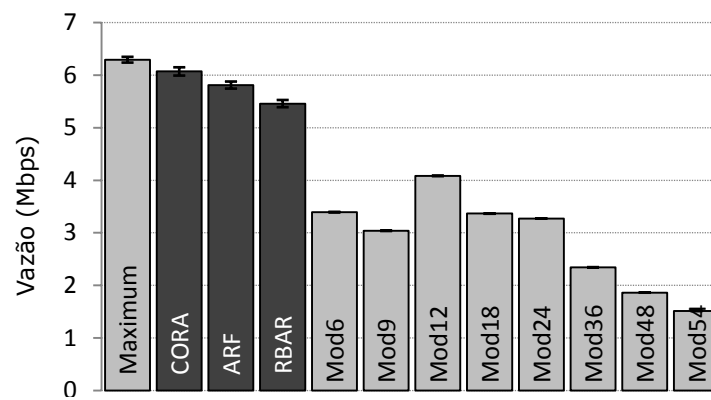


Figura 5.8: Vazão média dos algoritmos.

Em um segundo cenário de avaliação, foi considerado o caso de interferência na comunicação entre os nós M e E . Como ilustrado na Figura 5.9, foi acrescentado um par de nós estacionários e um fluxo CBR de $20Mbps$ entre eles. Este fluxo de interferência é um fluxo intermitente, isto é, intercala períodos regulares de transmissão e pausa.

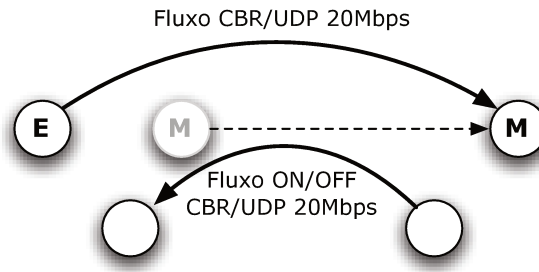


Figura 5.9: Cenário de rede simulado para avaliação do CORA.

Neste cenário, o mecanismo CORA produziu maior vazão média. De acordo com os resultados apresentados na Figura 5.10, o mecanismo proposto obteve um desempenho 4,9% melhor que o ARF e 13,1% melhor que o RBAR.

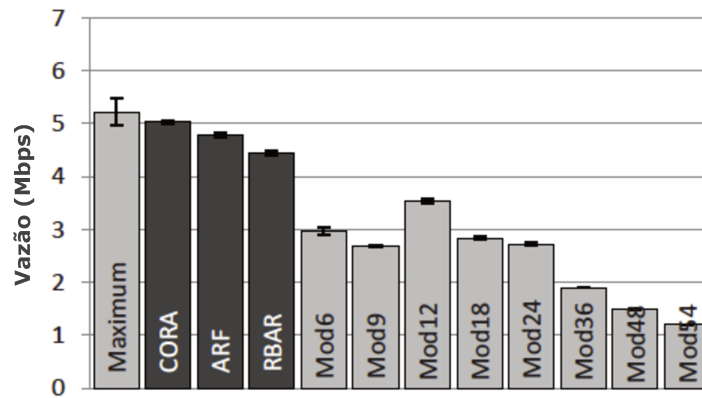


Figura 5.10: Vazão média.

Este trabalho foi realizado em conjunto com o estudante Luciano Chaves durante seu mestrado. Além dos resultados apresentados aqui, uma extensa avaliação do mecanismo CORA é apresentada em [14, 15].

5.3 Múltiplos Parâmetros

Em um terceiro estudo de caso, foram realizados experimentos para avaliar a eficácia do CogProt na reconfiguração dinâmica de múltiplos parâmetros. Neste estudo, foram definidos como parâmetros de interesse aqueles considerados nos estudos anteriores: o fator de incremento do TCP e a taxa de transmissão. Dessa forma, este estudo ilustra o caso de ajuste de parâmetros em camadas diferentes (cenário *inter-layer*).

O cenário de rede considerado neste estudo é ilustrado na Figura 5.11. Fluxos TCP são estabelecidos entre o nó móvel de origem S e o nó estacionário de destino D . Existe também um fluxo de interferência entre os nós C_1 e C_2 com o objetivo de gerar contenção no enlace entre os roteadores R_1 e R_2 , que estão no caminho entre S e D .

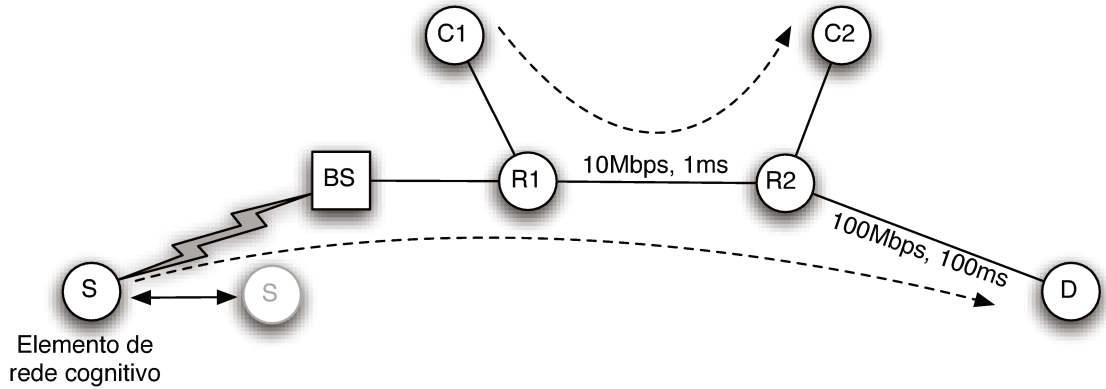


Figura 5.11: Cenário de rede para avaliação da abordagem cognitiva no ajuste dinâmico de múltiplos parâmetros.

Em relação aos parâmetros de interesse, considera-se $\alpha \in \{1, \dots, 4\}$ e também quatro possíveis taxas de transmissão $\{6, 18, 36 \text{ e } 54 \text{ Mbps}\}$. O valor padrão para o fator de incremento é 1 e para a taxa de transmissão é 6 Mbps . A métrica de desempenho é a vazão média do fluxo. O mecanismo mantém um histórico de desempenho para cada par de fator de incremento e taxa de transmissão na base de conhecimento.

Foram considerados 3 cenários de simulação, sendo que, para os dois primeiros, foram realizadas simulações com e sem o fluxo de interferência.

1. quando o nó origem S se mantém fixo próximo ($5m$) da estação base;
2. quando o nó origem S se mantém fixo distante ($50m$) da estação base;
3. quando o nó origem S se movimenta em relação à estação base.

Na Figura 5.12, são apresentados os resultados para cada par de fator de incremento e taxa de transmissão sem a presença de fluxo de interferência. Na ausência de fluxo de interferência, uma estratégia agressiva de crescimento da janela do TCP, com um alto valor para α é interessante para qualquer distância do nó de origem em relação à estação base. Pode-se observar que a vazão aumenta quando o valor de α aumenta. No entanto, o mesmo não vale para a taxa de transmissão. Para cada posição do nó de origem existe uma taxa mais adequada: 54Mbps quando S está próximo de BS e 18Mbps quando ele está distante da base.

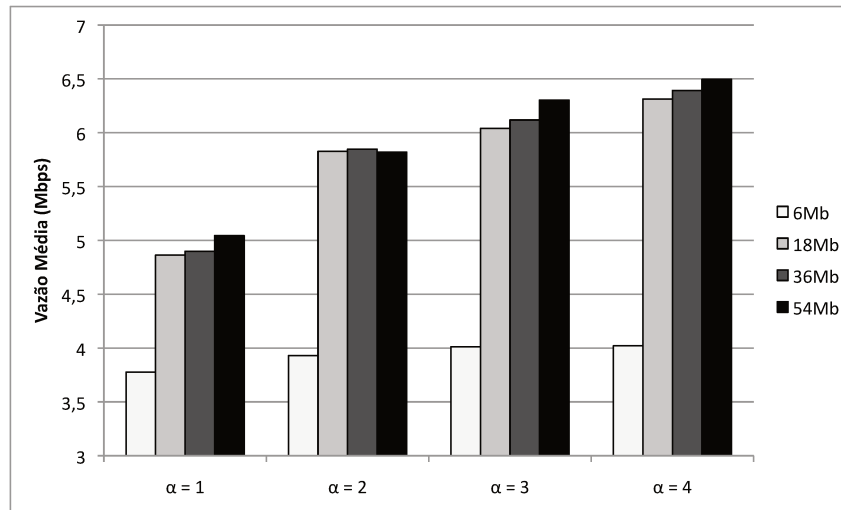
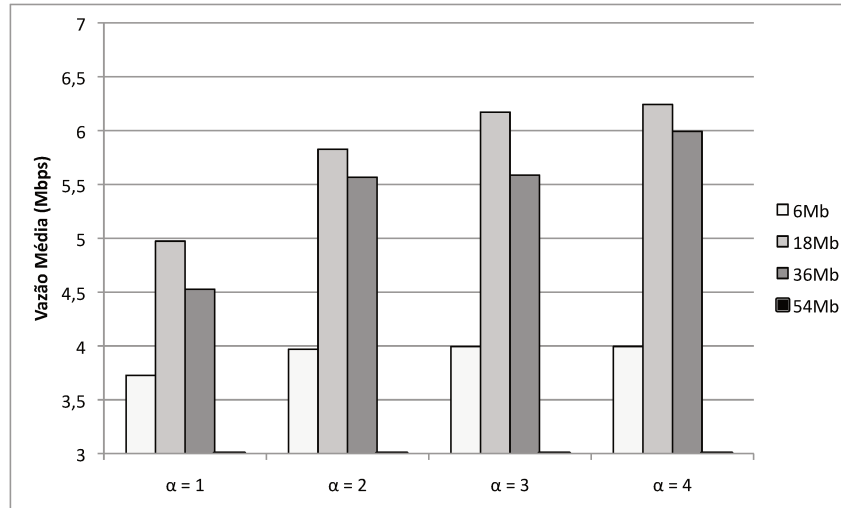
(a) Nó S próximo da estação base.(b) Nó S distante da estação base.

Figura 5.12: Vazão média sem tráfego de interferência para os casos onde S está próximo ou distante de BS .

A Figura 5.13 apresenta os resultados no caso da existência do fluxo de interferência. Pode-se observar que, independentemente da posição do nó S , uma estratégia conservadora de crescimento da janela TCP resulta em um melhor desempenho, ao contrário do que se observa nos resultados anteriores. Outra diferença é que mesmo quando o nó está próximo da estação base, existem casos nos quais taxas de transmissão mais baixas apresentam desempenho médio melhor que a taxa nominal máxima.

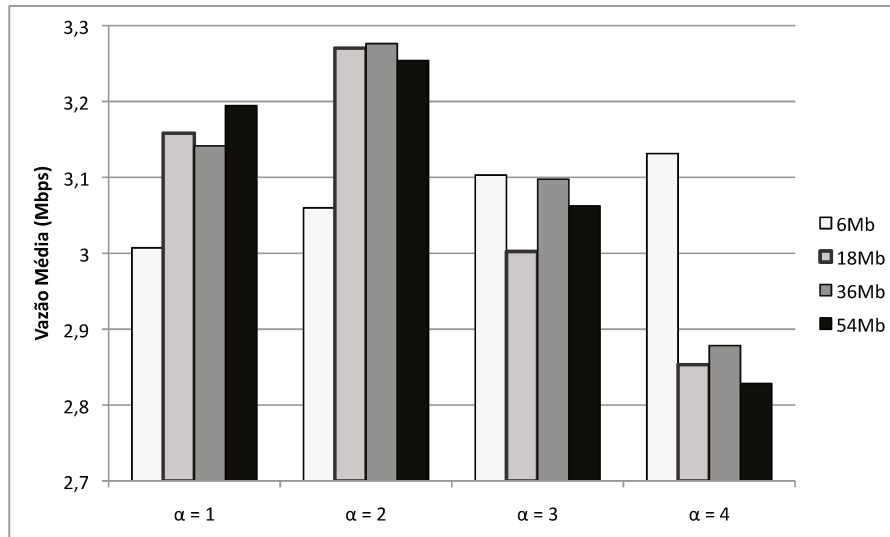
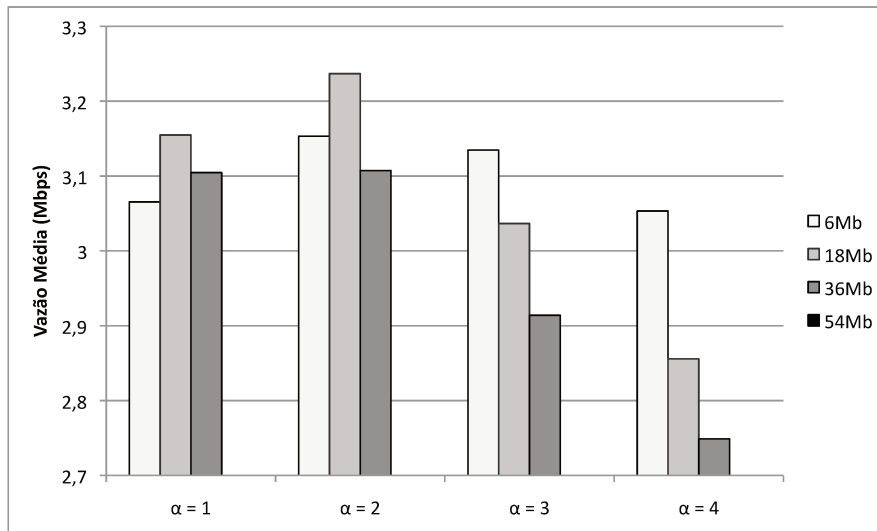
(a) Nó S próximo da estação base.(b) Nó S distante da estação base.

Figura 5.13: Vazão média com a presença de tráfego de interferência.

A partir dos resultados apresentados, não é possível identificar valores que sejam ótimos para ambos os cenários. Desta forma, fica evidente a necessidade do ajuste dinâmico desses parâmetros. O terceiro cenário de rede simulado representa um cenário heterogêneo, e portanto mais próximo dos cenários reais. Neste cenário, o nó S se movimenta afastando-se da posição inicial próxima à estação base. Além disso, foi ativado o fluxo de interferência, mas dessa vez de forma intermitente, para provocar alterações nas condições da rede durante a simulação.

Os resultados são apresentados na Figura 5.14, na qual são comparados o desempenho de cada par de fator de incremento e taxa, e o desempenho quando os parâmetros são ajustados dinamicamente por meio do CogProt. O mecanismo cognitivo provou ser eficaz em sustentar o melhor desempenho médio. Ele consegue adaptar os parâmetros em face das mudanças e evitar degradação de desempenho.

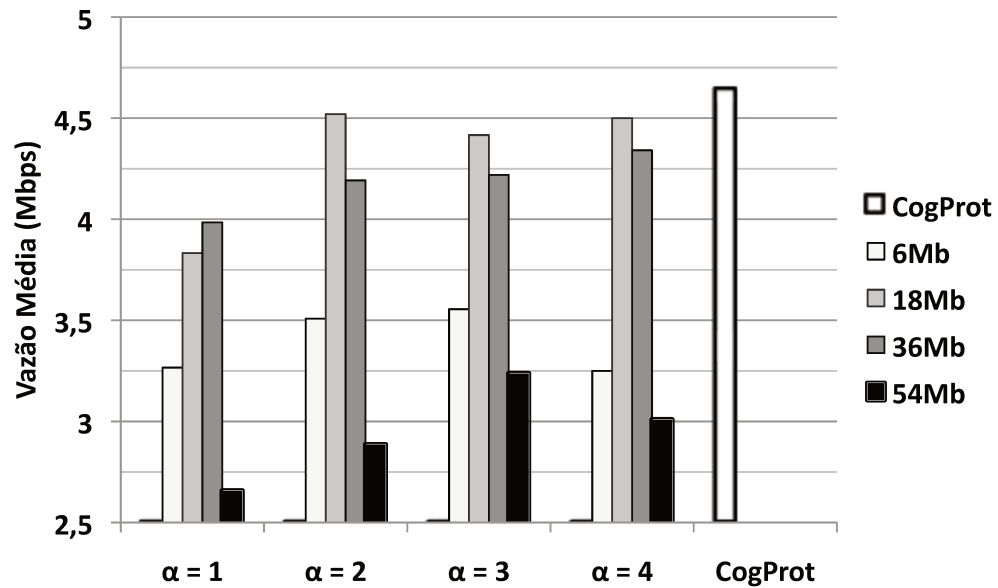


Figura 5.14: Vazão média em um cenário heterogêneo.

5.4 Implementação em Ambiente Real

Este último estudo de caso foi realizado com o objetivo de avaliar a viabilidade da implementação da abordagem proposta em elementos de rede reais. O mecanismo CogTCP foi implementado como descrito na Seção 5.1.

O sistema operacional Linux possui vários mecanismos de controle de congestionamento, como o Cubic [25] e o Reno. O sistema permite ao administrador escolher o mecanismo de controle e também oferece uma estrutura bem definida para implementação de novos mecanismos. Um novo mecanismo de controle de congestionamento é implementado como um módulo do núcleo do sistema operacional. O CogTCP foi implementado desta forma. O código fonte do CogTCP é apresentado no Apêndice A, assim como instruções para uso do módulo. Essa implementação estende o algoritmo AIMD do TCP Reno com o suporte à reconfiguração dinâmica do fator de incremento linear da janela de congestionamento.

O cenário de rede do experimento é ilustrado na Figura 5.15. O CogTCP foi usado em uma estação de trabalho, denominada S , no Laboratório de Redes na UNICAMP. Os fluxos foram estabelecidos com dois servidores: o servidor $D1$ na mesma rede local onde está S e o servidor $D2$ em uma rede na Universidade Federal do Mato Grosso do Sul (UFMS). Como observado, o caminho entre essas redes locais é estabelecido por roteadores de redes acadêmicas da Rede Nacional de Ensino e Pesquisa (RNP) e da Rede ANSP (*Academic Network at São Paulo*).

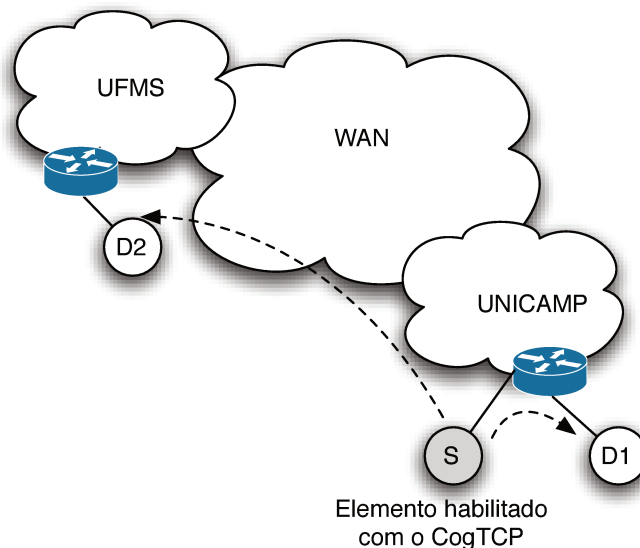


Figura 5.15: Cenário de rede do experimento com CogTCP.

Foi considerado o seguinte domínio para o parâmetro de interesse: $\alpha \in \{1, \dots, 5\}$. O valor padrão para o fator de incremento é $\alpha = 3$. A métrica de desempenho é a vazão média. Os parâmetros do ciclo de controle cognitivo foram determinados empiricamente como a seguir: $\tau = 1s$, $s = 0,75$ e $\sigma = 1,0$. Foram realizados experimentos para avaliar a vazão média de fluxos usando o TCP Reno tradicional com diferentes valores estáticos para α e fluxos usando o CogTCP. Este experimento é similar à simulação realizada com o CogTCP (como apresentado na Seção 5.1).

Neste experimento, consideramos dois cenários: fluxos na rede local na Unicamp (entre S e $D1$) e fluxos entre as redes locais das duas universidades ligadas pela rede nacional acadêmica (fluxos entre S e $D2$). Os fluxos TCP foram gerados com a ferramenta Iperf⁵. A própria ferramenta é capaz de medir a vazão média dos fluxos. A duração dos fluxos é de 100 segundos.

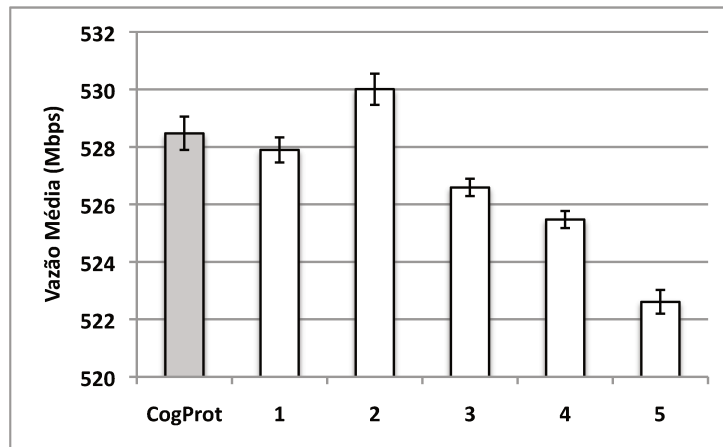
Os gráficos da Figura 5.16 mostram o desempenho na rede local e na rede acadêmica. Pode-se observar que o valor de $\alpha = 2$ apresenta o melhor desempenho na rede local, enquanto que $\alpha = 5$ é melhor para fluxos na rede acadêmica, ou seja, quando há fluxos para diferentes destinos em redes locais e de longa distância, não é possível escolher um valor ótimo global para α .

De fato, quando se calcula o desempenho médio considerando fluxos nas duas redes, tem-se que o CogTPC é mais eficiente, uma vez que ele é capaz de ajustar o valor de α durante o tempo de vida de um fluxo. Estes resultados são mostrados no gráfico da Figura 5.17. É importante notar que os resultados deste experimento confirmam o comportamento observado nas simulações com o CogTCP.

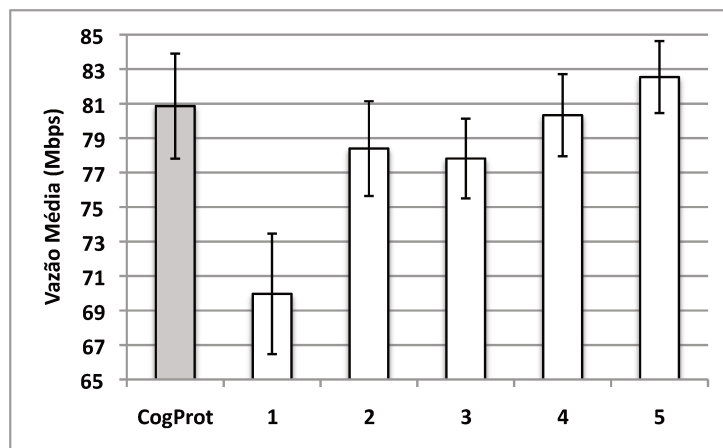
Nesta avaliação, estudou-se também o comportamento do CogTCP observando a frequência dos valores de α escolhidos aleatoriamente pelo mecanismo na tentativa de convergir para o valor ótimo. Os gráficos na Figura 5.18 a seguir ilustram a frequência de escolha para cada valor durante o tempo de vida de um fluxo TCP para cada um dos cenários de rede. Pode-se observar que os valores ótimos para cada cenário são aqueles escolhidos com maior frequência pelo algoritmo cognitivo em cada cenário, o que resulta no melhor desempenho médio quando considera-se um cenário realístico, ou seja, com fluxos em redes locais e de longa distância.

Verifica-se que o mecanismo pode ser implementado mesmo em elementos com recursos limitados, pois a implementação não exige alto poder de processamento, nem espaço de memória significativo.

⁵A ferramenta Iperf está disponível em <http://iperf.sourceforge.net>.



(a) Rede Local



(b) Rede Acadêmica

Figura 5.16: Desempenho nos diferentes cenários de rede.

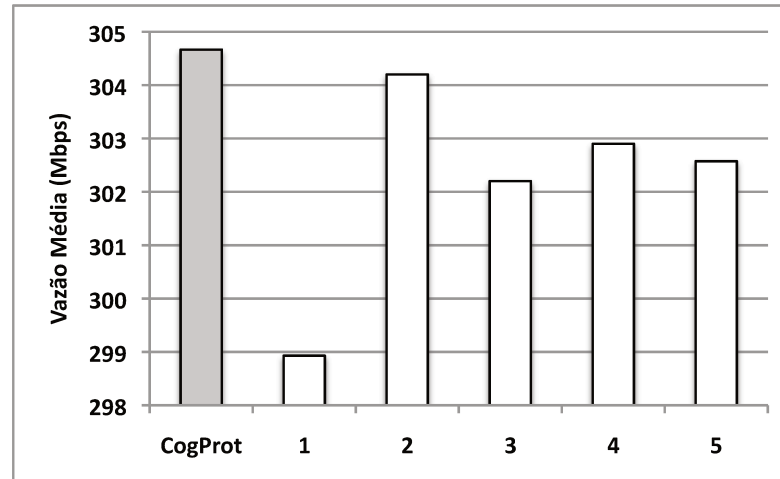
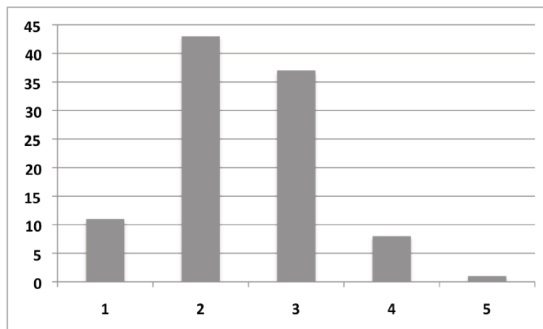
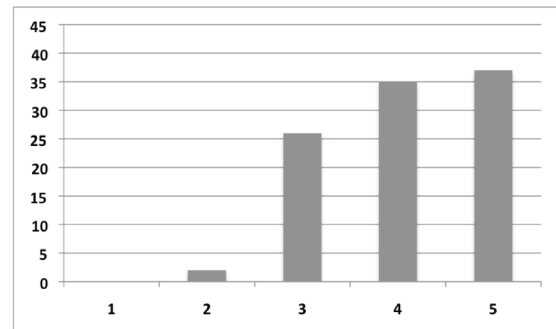


Figura 5.17: Desempenho médio com fluxos nas duas redes.



(a) Rede Local



(b) Rede Acadêmica

Figura 5.18: Freqüência de escolha dos valores de α .

Capítulo 6

Conclusão

Os sistemas de comunicação atuais são infraestruturas complexas e heterogêneas. A fim de se prover redes de alto desempenho com suporte a mobilidade e aplicações multimídia, avanços na área de comunicação de dados dependem, entre outros, do desenvolvimento de novas soluções de gerência de redes. Um dos principais requisitos é a capacidade de adaptação da rede para lidar com a heterogeneidade dos dispositivos e mudanças no estado da rede, como alterações na topologia e na demanda de tráfego.

Neste trabalho, foi proposta uma abordagem cognitiva para a reconfiguração dinâmica de protocolos de redes de comunicação. A abordagem define um arcabouço, denominado CogProt, que permite a implementação de mecanismos de auto-configuração. A solução proposta provê aos elementos de rede a capacidade de reajustar dinamicamente seus parâmetros de configuração a fim de evitar degradação de desempenho em função de mudanças nas condições da rede.

Inicialmente, apresentou-se um breve histórico das principais abordagens de gerência de rede. Em seguida, discutiu-se as principais características de diversas arquiteturas de auto-configuração. Então, a abordagem proposta foi apresentada em detalhes. Descreveu-se o processo de auto-configuração cognitivo que envolve, continuamente, analisar o estado da rede e então tomar decisões de ajuste de parâmetros de configuração do elemento de rede. Além disso, propôs-se um algoritmo genérico para implementação do processo de auto-configuração e discutiu-se diversos cenários de aplicação da solução proposta. Vários estudos de caso foram realizados para avaliação da abordagem CogProt. Primeiro, estudos baseados em simulações foram realizados para avaliar a aplicação do CogProt em protocolos da camada de transporte e de enlace. A reconfiguração dinâmica de parâmetros em diferentes camadas simultaneamente foi também avaliada através de simulação. Além disso, um experimento foi realizado para investigar a viabilidade e desempenho da solução proposta em um ambiente de rede real. O CogProt foi implementado como um módulo do Linux para ajustar o protocolo TCP dinamicamente.

O CogProt possui algumas limitações. Primeiro, a definição de seus parâmetros de controle (Tabela 4.2) é uma tarefa complexa e depende da aplicação específica. É preciso encontrar um equilíbrio adequado entre tempo de convergência e estabilidade, para definição dos valores do peso s e do desvio padrão (σ). Segundo, não é fácil determinar se o CogProt é adequado para a reconfiguração dinâmica de um parâmetro específico, pois isso exige estudo do impacto do parâmetro no desempenho da comunicação fim-a-fim.

Entre os pontos fortes da abordagem proposta pode-se destacar os seguintes. O processo cognitivo de auto-configuração é executado localmente no elemento de rede, o que caracteriza a natureza descentralizada do CogProt. Essa característica contribui para um maior grau de escalabilidade dos mecanismos projetados a partir desta abordagem. Por outro lado, é possível o uso de um serviço centralizado para oferecer informações adicionais que possam melhorar o desempenho do mecanismo. Este é o caso, por exemplo, de ajustes que dependem necessariamente de uma visão da rede com um todo.

O CogProt pode ser implantado de forma incremental em uma rede, uma vez que ele não altera as mensagens ou o funcionamento dos protocolos. Essa característica garante compatibilidade com a arquitetura TCP/IP e possibilidade dos elementos habilitados com o CogProt interagirem com elementos que não implementam a solução cognitiva. Além disso, esta solução pode ser aplicada na reconfiguração de uma ampla variedade de protocolos, em diferentes camadas da arquitetura de rede.

Os resultados mostram que a solução proposta é eficaz em melhorar o desempenho médio da rede por meio do ajuste dinâmico dos parâmetros de interesse a um custo relativamente baixo. Isto porque os procedimentos de análise e decisão são simples e não requerem esquema de representação de informação com alto nível de abstração. A solução proposta não demanda alto poder computacional ou capacidade de memória, de forma que ela pode ser implementada em equipamentos com recursos limitados.

Este trabalho de pesquisa gerou os seguintes artigos:

- D. Kliazovich, N. Malheiros, N. L. S. da Fonseca, F. Granelli, e E. Madeira. CogProt: A framework for cognitive configuration and optimization of communication protocols. *Mobile Lightweight Wireless Systems, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol 45, 280–291. Springer, 2010.
- N. Malheiros, D. Kliazovich, F. Granelli, E. Madeira, e N. L. S. da Fonseca. A cognitive approach for cross-layer performance management. *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM'10)*, 2010.
- Malheiros, N.; Lima, J.; Chaves, L.; Madeira, E.; da Fonseca, N. L. S.; e Garcia, I. Uma Abordagem Cognitiva para Redes Autoconfiguráveis. *II Workshop de Sistemas Distribuídos Autônomicos. Anais do XXVIII SBRC 2012*.

- N. Malheiros, D. Kliazovich, F. Granelli, E. Madeira, e N. L. S. da Fonseca. A Cognitive Approach for Self-Configuration of Communication Protocols. SUBMETIDO para IEEE Transactions on Network and Service Management.

6.1 Trabalhos Futuros

A seguir, são apresentados tópicos para trabalhos futuros:

Novas Aplicações A abordagem proposta é genérica o bastante para ser aplicada em uma variedade de cenários. Neste trabalho, apresentou-se a aplicação da proposta na auto-configuração do TCP e no controle da taxa de transmissão em redes 802.11. O arcabouço proposto já foi também aplicado com sucesso para auto-configuração de outros parâmetros de redes sem fio [39]. Pode-se explorar a aplicação dessa abordagem em outros cenários, por exemplo, no controle de potência de transmissão em redes ad hoc.

Modelo Analítico Foram realizadas algumas atividades de avaliação de desempenho por meio de simulações e da implementação da solução proposta em um ambiente de rede real. Um passo importante seria conceber um modelo analítico através do qual fosse possível determinar propriedades do mecanismo proposto, por exemplo, o tempo de convergência para uma aplicação específica, assim como os valores ótimos para as variáveis de controle do CogProt para um cenário de aplicação específico.

Parametrização do CogProt Ajustar as variáveis de controle do CogProt durante a operação da rede pode melhorar o desempenho do sistema. O valor do desvio padrão usado na distribuição normal tem um impacto direto no tempo de convergência e na estabilidade do mecanismo. Uma contribuição importante seria investigar uma estratégia para ajuste dinâmico do desvio padrão, assim como para as outras variáveis de controle do CogProt, a saber, o intervalo de operação do ciclo de controle (τ) e o peso s utilizado na fase de análise. Ou até mesmo, verificar a possibilidade de desvincular o intervalo do ciclo de controle do intervalo de monitoramento do desempenho na fase de análise em casos específicos.

Protocolos de Gerência Ainda é preciso investigar mecanismos de comunicação para compartilhamento de informações de gerência entre elementos cognitivos. O compartilhamento dessas informações pode ser feito de forma descentralizada ou mediado por um serviço na rede. Essas informações podem aumentar a eficiência dos mecanismos de auto-configuração.

Referências Bibliográficas

- [1] A. Afanasyev, N. Tilley, P. Reiher, and L. Kleinrock. Host-to-host congestion control for tcp. *IEEE Communications Surveys & Tutorials*, 12(3):304–342, 2010.
- [2] E. Ancillotti, R. Bruno, and M. Conti. Experimentation and performance evaluation of rate adaptation algorithms in wireless mesh networks. In *Proc. of the 5th ACM symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks (PE-WASUN’08)*, pages 7–14, 2008.
- [3] J. C. Bicket. Bit-rate selection in wireless networks. Master’s thesis, Massachusetts Institute of Technology, 2005.
- [4] A. Bieszczad, B. Pagurek, and T. White. Mobile agents for network management. *IEEE Communications Surveys & Tutorials*, 1(1):2–9, 1998.
- [5] Andreas Binzenhöfer, Kurt Tutschku, Björn Graben, Markus Fiedler, and Patrik Arlos. A P2P-Based Framework for Distributed Network Management. In Matteo Cesana and Luigi Fratta, editors, *Wireless Systems and Network Architectures in Next Generation Internet*, volume 3883 of *Lecture Notes in Computer Science*, pages 198–210. Springer Berlin Heidelberg, 2006.
- [6] G. Bouabene, C. Jelger, C. Tschudin, S. Schmid, A. Keller, and M. May. The Autonomic Network Architecture (ANA). *IEEE Journal on Selected Areas in Communications*, 28(1):4–14, 2010.
- [7] M. Bouet, G. Nguengang, V. Conan, A. Kousaridas, P. Spapis, and N. Alonistioti. Embedding cognition in wireless network management: an experimental perspective. *IEEE Communications Magazine*, 50(12):150–160, 2012.
- [8] R. Boutaba and I. Aib. Policy-based Management: A Historical Perspective. *Journal of Network and Systems Management*, 15(4):447–480, 2007.

- [9] R. Boutaba, J. Xiao, and Q. Zhang. Toward autonomic networks: Knowledge management and self-stabilization. In Y. Zhang, L. T. Yang, and M. K. Denko, editors, *Autonomic Computing and Networking*, pages 239–260. Springer, 2009.
- [10] T. Braga, F. Silva, L. Ruiz, and H. Assunção. Redes autonômicas. In *Anais dos Minicursos do XXIV Simpósio Brasileiro de Redes de Computadores*, pages 159–208. SBC, 2006.
- [11] J. D. Case, M. Fedor, M. L. Schoffstall, and J. Davin. A Simple Network Management Protocol. IETF RFC 1067, 1988.
- [12] R. Chadha. A Cautionary Note About Policy Conflict Resolution. In *IEEE Military Communications Conference (MILCOM'06)*, pages 1–8, 2006.
- [13] R. Chadha, H. Cheng, Y. Cheng, J. Chiang, A. Ghetie, G. Levin, and H. Tanna. Policy-based mobile ad hoc network management. In *Proc. of the Fifth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'04)*, pages 35–44, 2004.
- [14] L. Chaves, N. Malheiros, E. Madeira, I. Garcia, and D. Kliazovich. A cognitive mechanism for rate adaptation in wireless networks. In J. Strassner and Y. Ghamri-Doudane, editors, *Modelling Autonomic Communications Environments*, volume 5844 of *Lecture Notes in Computer Science*, pages 58–71. Springer, 2009.
- [15] L. J. Chaves. Um Mecanismo Cognitivo para Adaptação Automática da Taxa de Transmissão em Redes IEEE 802.11. Master's thesis, Universidade Estadual de Campinas, 2010.
- [16] D. Croce, M. Mellia, and E. Leonardi. The quest for bandwidth estimation techniques for large-scale distributed systems. *SIGMETRICS Perform. Eval. Rev.*, 37(3):20–25, 2010.
- [17] S. Davy, B. Jennings, and J. Strassner. On harnessing information models and ontologies for policy conflict analysis. In *IFIP/IEEE International Symposium on Integrated Network Management (IM'09)*, pages 821–826, 2009.
- [18] P. Demestichas, V. Stavroulaki, D. Boscovic, A. Lee, and J. Strassner. m@ANGEL: autonomic management platform for seamless cognitive connectivity to the mobile Internet. *IEEE Communications Magazine*, 44(6):118–127, 2006.
- [19] X. Deng, S. Yi, G. Kesidis, and C. R. Das. A control theoretic approach for designing adaptive AQM schemes. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM'03)*, volume 5, pages 2947–2951, 2003.

- [20] S. Dobson, S. Denazis, A. Fernández, D. Gaïti, E. Gelenbe, F. Massacci, P. Nixon, F. Saffre, N. Schmidt, and F. Zambonelli. A survey of autonomic communications. *ACM Trans. Auton. Adapt. Syst.*, 1(2):223–259, 2006.
- [21] M. Fiedler, T. Hossfeld, and P. Tran-Gia. A generic quantitative relationship between Quality of Experience and Quality of Service. *IEEE Network*, 24(2):36–41, 2010.
- [22] T. Fioreze, L. Z. Granville, M. J. Almeida, and L. R. Tarouco. Comparing Web services with SNMP in a management by delegation environment. In *Proceedings of the 9th IFIP/IEEE International Symposium on Integrated Network Management*, pages 601–614, 2005.
- [23] C. Fortuna and M. Mohorcic. Trends in the Development of Communication Networks: Cognitive Networks. *Computer Networks*, 53(9):1354–1376, 2009.
- [24] G. Goldszmidt and Y. Yemini. Delegated agents for network management. *IEEE Communications Magazine*, 36(3):66–70, 1998.
- [25] S. Ha, I. Rhee, and L. Xu. CUBIC: a new TCP-friendly high-speed TCP variant. *SIGOPS Oper. Syst. Rev.*, 42(5):64–74, 2008.
- [26] S. Haykin. Cognitive radio: brain-empowered wireless communications. *IEEE Journal on Selected Areas in Communications*, 23(2):201–220, 2005.
- [27] T. Heikkinen. A potential game approach to distributed power control and scheduling. *Computer Networks*, 50(13):2295–2311, 2006.
- [28] G. Holland, N. Vaidya, and P. Bahl. A rate-adaptive mac protocol for multi-hop wireless networks. In *Proc. of the 7th ACM International Conference on Mobile Computing and Networking (MobiCom’01)*, pages 236–251, 2001.
- [29] Y. Huang, S. Mao, and S. F. Midkiff. A Control-Theoretic Approach to Rate Control for Streaming Videos. *IEEE Transactions on Multimedia*, 11(6):1072–1081, 2009.
- [30] M. C. Huebscher and J. A. McCann. A survey of autonomic computing – degrees, models, and applications. *ACM Comput. Surv.*, 40(3):7:1–7:28, 2008.
- [31] B. Jennings, R. Brennan, W. Donnelly, S. N. Foley, D. Lewis, D. O’Sullivan, J. Strasner, and S. van der Meer. Challenges for federated, autonomic network management in the future internet. In *IFIP/IEEE International Symposium on Integrated Network Management-Workshops (IM’09)*, pages 87–92, 2009.

- [32] B. Jennings, S. van der Meer, S. Balasubramaniam, D. Botvich, M. O Foghlu, W. Donnelly, and J. Strassner. Towards autonomic management of communications networks. *IEEE Communications Magazine*, 45(10):112–121, 2007.
- [33] Z. Ji and K. J. R. Liu. Dynamic Spectrum Sharing: A Game Theoretical Overview. *IEEE Communications Magazine*, 45(5):88–94, 2007.
- [34] A. Kamerman and L. Monteban. WaveLAN-II: A High-Performance Wireless LAN for the Unlicensed Band. *Bell Labs Technical Journal*, 2(3):118–133, 1997.
- [35] Meriem Kassar, Brigitte Kervella, and Guy Pujolle. An overview of vertical handover decision strategies in heterogeneous wireless networks. *Computer Communications*, 31(10):2607 – 2620, 2008.
- [36] J. O. Kephart. Research challenges of autonomic computing. In *Proc. of the 27th International Conference on Software Engineering (ICSE'05)*, pages 15–22, 2005.
- [37] J. O. Kephart and D. M. Chess. The vision of autonomic computing. *Computer*, 36(1):41–50, 2003.
- [38] D. Kliazovich, F. Granelli, and N. L. S. Da Fonseca. Architectures and Cross-Layer Design for Cognitive Networks. In *Handbook on Sensor Networks*, pages 3–24. World Scientific Publishing, 2010.
- [39] D. Kliazovich, J. Lima, N. L. S. da Fonseca, F. Granelli, and E. Madeira. Cognitive link layer for wireless local area networks. In *IEEE Latin-American Conference on Communications (LATINCOM '09)*, pages 1–6, 2009.
- [40] J. F. Kurose and K. W. Ross. *Computer networking: a top-down approach*. Addison Wesley, 5th edition, 2010.
- [41] S. M. Lake. Cognitive networking with software programmable intelligent networks for wireless and wireline critical communications. In *IEEE Military Communications Conference (MILCOM'05)*, volume 3, pages 1693–1699, 2005.
- [42] A. B. MacKenzie and S. B. Wicker. Game theory and the design of self-configuring, adaptive wireless networks. *IEEE Communications Magazine*, 39(11):126–131, 2001.
- [43] B. S. Manoj, R. R. Rao, and M. Zorzi. Cognet: a cognitive complete knowledge network system. *IEEE Wireless Communications*, 15(6):81–88, 2008.
- [44] Z. Movahedi, M. Ayari, R. Langar, and G. Pujolle. A Survey of Autonomic Network Architectures and Evaluation Criteria. *IEEE Communications Surveys & Tutorials*, 14(2):464 –490, 2012.

- [45] A. Panisson, D.M. da Rosa, C. Melchior, L.Z. Granville, M.J.B. Almeida, and L.M. Rockenbach Tarouco. Designing the Architecture of P2P-Based Network Management Systems. In *Proc. of the 11th IEEE Symposium on Computers and Communications (ISCC)*, pages 69–75, 2006.
- [46] Jong-Tae Park, Jong-Wook Baek, and J. Woon-Ki Hong. Management of service level agreements for multimedia internet service using a utility model. *Communications Magazine, IEEE*, 39(5):100–106, 2001.
- [47] G. M. Perez, A. F. G. Skarmeta, S. Zeber, J. Spagnolo, and T. Symchych. Dynamic Policy-Based Network Management for a Secure Coalition Environment. *IEEE Communications Magazine*, 44(11):58–64, 2006.
- [48] N. Samaan and A. Karmouch. Towards autonomic network management: an analysis of current and future research directions. *IEEE Communications Surveys & Tutorials*, 11(3):22–36, 2009.
- [49] V. Srivastava and M. Motani. Cross-layer design: a survey and the road ahead. *IEEE Communications Magazine*, 43(12):112–119, 2005.
- [50] P. Sutton, L. E. Doyle, and K. E. Nolan. A Reconfigurable Platform for Cognitive Networks. In *Proceedings of the 1st International Conference on Cognitive Radio Oriented Wireless Networks and Communications*, pages 1–5, 2006.
- [51] Cisco Systems. CiscoWorks Network Compliance Manager 1.7: Product Data Sheet, 2011. Disponível em: http://www.cisco.com/en/US/prod/collateral/netmgts/ps6911/ps6923/data_sheet_c78-658607.pdf. Acessado em Julho de 2012.
- [52] R. W. Thomas, D. H. Friend, L. A. Dasilva, and A. B. Mackenzie. Cognitive networks: adaptation and learning to achieve end-to-end performance objectives. *IEEE Communications Magazine*, 44(12):51–57, 2006.
- [53] D. C. Verma. Simplifying network administration using policy-based management. *IEEE Network*, 16(2):20–26, 2002.

Apêndice A

Implementação do CogTCP

Neste trabalho, o mecanismo CogTCP foi implementado como um módulo do núcleo do sistema operacional Linux. A distribuição “Ubuntu Linux” foi a utilizada para realização dos experimentos. O código fonte¹ do CogTCP é apresentado na Seção A.1. Uma vez carregado no sistema, esse módulo se apresenta como um mecanismo de controle de congestionamento do TCP, entre os já disponíveis no sistema. A seguir, são apresentados alguns comandos importantes para uso deste módulo.

- Para verificar quais os mecanismos disponíveis no sistema:
`$ sysctl net.ipv4.tcp_available_congestion_control`
- Para verificar qual mecanismo está habilitado:
`$ sysctl net.ipv4.tcp_congestion_control`
- Para compilar o módulo, pode-se utilizar o *Makefile* cujo código fonte é apresentado na Seção A.2. Considere que ambos os arquivos fontes estão em um diretório denominado “cogtcp”, então o módulo pode ser compilado da seguinte forma:
`$ cd cogtcp`
`$ make`
- Para carregar e habilitar o módulo CogTCP, devem ser executados os comandos a seguir. Esses comandos exigem privilégio de administrador do sistema. As conexões TCP já iniciadas antes da habilitação do módulo não sofrem qualquer alteração. Apenas as novas conexões utilizarão o módulo habilitado para controle de congestionamento.
`$ insmod cogtcp.ko`
`$ sysctl -w net.ipv4.tcp_congestion_control=cogtcp`

¹O código fonte está disponível no seguinte endereço: <http://www.lrc.ic.unicamp.br/cogprot>

- O comando a seguir remove o módulo CogTCP do núcleo do sistema. Neste caso, também se exige privilégio de administrador. O módulo não pode ser removido, se estiver habilitado. Assim, antes de tentar remover o módulo é necessário habilitar outro mecanismo de controle de congestionamento.

```
$ rmmod cogtcp
```

A.1 Código Fonte do CogTCP

```

1 #include <linux/module.h>
2 #include <linux/timer.h>
3 #include <net/tcp.h>
4
5 // we consider that increase factor is an integer in {1..5}
6
7 /*
8  * Phases of the quality feedback loop
9  * Analysis - Decision - Action
10 */
11 void analysis (unsigned long data);
12 void decision (struct sock *sk);
13 void action (struct sock *sk);
14
15 // interval for the quality feedback loop (100 miliseconds)
16 static unsigned int loop_interval = (100 * HZ) / 1000;
17
18 struct cogtcp {
19     unsigned int ai_factor;           // aditive increase factor
20     unsigned int performance_info_base[5]; //avg throughput for every factor
21     struct timer_list *feedback;      // timer to implement the feedback loop
22     unsigned int previous_bytes;      // auxiliar var to compute performance
23     unsigned int best_ai_factor;      // the current optimal ai_factor
24 };
25
26 // calculate performance for current factor
27 void analysis (unsigned long data)
28 {
29     int pos;
30     unsigned int bytes;
31
32     // get cogtcp parameters
33     struct sock *sk = (struct sock*)data;
34     struct tcp_sock *tp = tcp_sk(sk);
35     struct cogtcp *cog_pars = inet_csk_ca(sk);

```

```

36
37  /* measure sent bytes since last analysis */
38  bytes = (unsigned int)tp->snd_nxt - cog_pars->previous_bytes;
39  cog_pars->previous_bytes = (unsigned int)tp->snd_nxt;
40
41  // exponential moving average to estimate performance
42  /* avg_throuhput = 7/8 * avg_throughput + 1/8 * measured (we consider
   weight = 0.125 = 1/8) */
43  if (bytes != 0) {
44      // avg performance for ai_factor x is sotred on position x-1
45      pos = cog_pars->ai_factor - 1;
46      cog_pars->performance_info_base[pos] = 7 * (cog_pars->
          performance_info_base[pos] >> 3) + (bytes >> 3);
47  }
48
49  decision(sk);
50  action(sk);
51
52  cog_pars->feedback->expires = jiffies + loop_interval;
53  add_timer(cog_pars->feedback);
54 }
55
56 // decision - look up for factor with best average performance
57 void decision (struct sock *sk)
58 {
59     unsigned int best_factor_pos = 0;
60     unsigned int best_performance = 0;
61     unsigned int i;
62
63     struct cogtcp *cog_pars = inet_csk_ca(sk);
64
65     for(i = 0; i < 5; i++) {
66         if (cog_pars->performance_info_base[i] > best_performance) {
67             best_factor_pos = i;
68             best_performance = cog_pars->performance_info_base[i];
69         }
70     }
71     cog_pars->best_ai_factor = best_factor_pos+1;
72 }
73
74
75 // action -> assign TCP window increase factor to random factor
76 void action (struct sock *sk)
77 {
78     struct cogtcp *cog_pars = inet_csk_ca(sk);
79     u32 rand;

```

```

80     int i;
81     unsigned int random_factor;
82     int delta;
83
84     /* normal distribution (based on convolution method, refer to "Raj Jain,
85        the art of computer systems performance analysis, page 494")*/
86     delta = 0;
87     for (i = 0; i < 12; i++) {
88         get_random_bytes(&rand, 4);
89         delta += (int)(rand & 0x00FFFFFF);
90     }
91     delta = (3 * (delta - 0x5FFFFFFA)) / 2;    // deviation = 1.5 = 3/2; mean =
92         0;
93     for (i = 1; i <= 5; i++) {
94         if ( delta > (0x5FFFFFFA - (i* 0x26666664))) {
95             delta = 3 - i;
96             break;
97         }
98     } // here we get delta <= (normal) random integer in [-2,2]
99
100    // we consider "best_ai_factor" is the mean of the normal distribution
101    random_factor = cog_pars->best_ai_factor + delta;
102
103    // must assure ai_factor in {1..5}
104    if (random_factor < 1)
105        cog_pars->ai_factor = 1;
106    else
107        if (random_factor > 5)
108            cog_pars->ai_factor = 5;
109        else
110            cog_pars->ai_factor = (unsigned int)random_factor;
111    }
112
113    static void cogtcp_init(struct sock *sk)
114    {
115        struct tcp_sock *tp = tcp_sk(sk);
116        struct cogtcp *cog_pars = inet_csk_ca(sk);
117        int i;
118
119        printk(KERN_INFO "_initing_cogtcp\n");
120
121        cog_pars->ai_factor = 1;    // default increase factor
122        cog_pars->previous_bytes = (unsigned int)tp->snd_nxt;
123        cog_pars->best_ai_factor = 1;
124    }

```

```

125     for(i = 0; i < 5; i++) {
126         cog_pars->performance_info_base[i] = 0;
127     }
128     cog_pars->feedback = kmalloc(sizeof(struct timer_list), GFP_ATOMIC
129                                );
129     if ( !cog_pars->feedback ) {
130         printk(KERN_INFO "cogtcp-init: _memory_allacation_failed.");
131         return;
132     }
133     init_timer(cog_pars->feedback);
134     cog_pars->feedback->expires = jiffies + loop_interval;
135     cog_pars->feedback->data = (unsigned long)sk;
136     cog_pars->feedback->function = analysis;
137     add_timer(cog_pars->feedback);
138 }
139
140 static void cogtcp_release(struct sock *sk) {
141     struct cogtcp *cog_pars = inet_csk_ca(sk);
142     del_timer(cog_pars->feedback);
143     kfree(cog_pars->feedback);
144 }
145
146 static void cogtcp_cong_avoid(struct sock *sk, u32 ack, u32 in_flight)
147 {
148     struct tcp_sock *tp = tcp_sk(sk);
149     struct cogtcp *cog_pars = inet_csk_ca(sk);
150
151     if (!tcp_is_cwnd_limited(sk, in_flight))
152         return;
153
154     /* In "safe" area, increase. */
155     if (tp->snd_cwnd <= tp->snd_ssthresh)
156         tcp_slow_start(tp);
157
158     /* In dangerous area, increase slowly. */
159     else {
160         if (tp->snd_cwnd_cnt >= tp->snd_cwnd) {
161             if ((tp->snd_cwnd+cog_pars->ai_factor) <= tp->snd_cwnd.clamp)
162                 tp->snd_cwnd += cog_pars->ai_factor;
163             tp->snd_cwnd_cnt = 0;
164         } else {
165             tp->snd_cwnd_cnt++;
166         }
167     }
168 }
169

```

```

170 static struct tcp_congestion_ops tcp_cogtcp = {
171     .init          = cogtcp_init ,
172     .release       = cogtcp_release ,
173     .sssthresh     = tcp_reno_sssthresh ,
174     .cong_avoid    = cogtcp_cong_avoid ,
175     .min_cwnd      = tcp_reno_min_cwnd ,
176     .owner         = THIS_MODULE,
177     .name          = "cogtcp"
178 };
179
180 static int __init cogtcp_register(void)
181 {
182     BUILD_BUG_ON(sizeof(struct cogtcp) > ICSK_CA_PRIV_SIZE);
183     return tcp_register_congestion_control(&tcp_cogtcp);
184 }
185
186 static void __exit cogtcp_unregister(void)
187 {
188     tcp_unregister_congestion_control(&tcp_cogtcp);
189 }
190
191 module_init(cogtcp_register);
192 module_exit(cogtcp_unregister);
193
194 MODULE_AUTHOR("Neumar Malheiros");
195 MODULE_LICENSE("GPL");
196 MODULE_DESCRIPTION("Cognitive_TCP");
197 MODULE_VERSION("0.5");

```

A.2 Código Fonte do Makefile

```

1 obj-m += cogtcp.o
2 all:
3     make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
4 clean:
5     make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean

```